Rectified Factor Networks Supplement

Djork-Arné Clevert, Andreas Mayr, Thomas Unterthiner and Sepp Hochreiter
Institute of Bioinformatics, Johannes Kepler University, Linz, Austria
{okko,mayr,unterthiner,hochreit}@bioinf.jku.at

Contents

List of Theorems

List of Algorithms

1 Introduction

This supplement contains additional information complementing the main manuscript and is structured as follows: First, the rectified factor network (RFN) learning algorithm with E- and M-step updates, weight decay and dropout regularization is given in Section 2. In Section 3, we proof that the (RFN) learning algorithm is a "generalized alternating minimization" (GAM) algorithm and converges to a solution that maximizes the RFN objective. The correctness of the RFN algorithm is proofed in Section 4. Section 5 describes the maximum likelihood factor analysis model and the model selection by the EM-algorithm. The RFN objective, which has to be maximized, is described in Section 6. Next, RFN's GAM algorithm via gradient descent both in the M-step and the E-step is reported in the Section 7. The following sections 8 and 9 describe the gradient-based M- and E-step, respectively. In Section 10, we describe how the RFNs sparseness can be controlled by a Gaussian prior. Additional information on the selected hyperparameters of the benchmark methods is given in Section 11. The sections 12 and 13 describe the data generation of the benchmark datasets and report the results for three different experimental settings, namely for extracting 50 (undercomplete), 100 (complete) or 150 (overcomplete) factors / hidden units. In Section 14 describes experiments, that we have done to assess the performance of RFN *first layer* pretraining on *CIFAR-10* and *CIFAR-100* for three deep convolutional network architectures: (i) the AlexNet Ciresan et al. [2012], Krizhevsky et al. [2012], (ii) Deeply Supervised Networks (DSN) Lee et al. [2014], and (iii) our 5-Convolution-Network-In-Network (5C-NIN). Finally, Section 15 provides running times for RFN's projected Newton step and for solving a quadratic program.

2 Rectified Factor Network (RFN) Algorithms

Algorithm 1 is the rectified factor network (RFN) learning algorithm. The RFN algorithm calls Algorithm 2 to project the posterior probability p_i onto the family of rectified and normalized variational distributions Q_i. Algorithm 2 guarantees an improvement of the E-step objective $O = \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(Q_i \parallel p_i)$. Projection Algorithm 2 relies on different projections, where a more complicated projection is tried if a simpler one failed to improve the E-step objective. If all following Newton-based gradient projection methods fail to decrease the E-step objective, then projection Algorithm 2 falls back to gradient projection methods. First the equality constraints are solved and inserted into the objective. Thereafter, the constraints are convex and gradient projection methods are applied. This approach is called "generalized reduced gradient method" Abadie and Carpentier [1969], which is our preferred alternative method. If this method fails, then Rosen's gradient projection method Rosen [1961] is used. Finally, the method of Haug and Arora Haug and Arora [1979] is used.

First we consider Newton-based projection methods, which are used by Algorithm 2. Algorithm 4 performs a simple projection, which is the projected Newton method with learning rate set to one. This projection is very fast and ideally suited to be performed on GPUs for RFNs with many coding units. Algorithm 3 is the fast and simple projection without normalization even simpler than Algorithm 4. Algorithm 5 generalizes Algorithm 4 by introducing step sizes λ and γ. The step size λ scales the gradient step, while γ scales the difference between to old projection and the new projection. For both λ and γ annealing steps, that is, learning rate decay is used to find an appropriate update.

If these Newton-based update rules do not work, then Algorithm 6 is used. Algorithm 6 performs a scaled projection with a reduced Hessian matrix $\boldsymbol{H}$ instead of the full Hessian $\boldsymbol{\Sigma}_p^{-1}$. For computing $\boldsymbol{H}$ an ϵ-active set is determined, which consists of all j with $\mu_j \leq \epsilon$. The reduced matrix $\boldsymbol{H}$ is the Hessian $\boldsymbol{\Sigma}_p^{-1}$ with ϵ-active columns and rows j fixed to unit vector $\boldsymbol{e}_j$.

The RFN algorithm allows regularization of the parameters $\boldsymbol{W}$ and $\boldsymbol{\Psi}$ (off-diagonal elements) by weight decay. Priors on the parameters can be introduced. If the priors are convex functions, then convergence of the RFN algorithm is still ensured. The weight decay Algorithm 7 can optionally be used after the M-step of Algorithm 1. Coding units can be regularized by dropout. However dropout is not covered by the convergence proof for the RFN algorithm. The dropout Algorithm 8 is applied during the projection between rectifying and normalization. Methods like mini-batches or other stochastic gradient methods are not covered by the convergence proof for the RFN algorithm. However, in Gunawardana and Byrne [2005] it is shown how to generalize the GAM convergence

proof to mini-batches as it is shown for the incremental EM algorithm. Dropout and other stochastic gradient methods can be show to converge similar to mini-batches.

Algorithm 1 Rectified Factor Network

Input
for $1 \leq i \leq n$: $\boldsymbol{v}_i \in \mathbb{R}^m$,
number of coding units l
Hyper-Parameters
$\Psi_{\min}, W_{\max}, \eta_\Psi, \eta_W, \rho, \tau, 1 < \eta \leq 1$
Initialization
$\boldsymbol{\Psi} = \tau \boldsymbol{I}$, $\boldsymbol{W}$ element-wise random in $[-\rho, \rho]$,
$\boldsymbol{C} = \frac{1}{n} \sum_{k=1}^{n} \boldsymbol{v}_k \, \boldsymbol{v}_k^T$, STOP=false
Main
while STOP=false **do**
——E-step1——
for all $1 \leq i \leq n$ **do**
$(\boldsymbol{\mu}_p)_i = \left(\boldsymbol{I} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \, \boldsymbol{v}_i$
end for
$\boldsymbol{\Sigma} = \left(\boldsymbol{I} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1}$
——Projection——
perform projection of $(\boldsymbol{\mu}_p)_i$ onto the feasible set by Algorithm 2 giving $\boldsymbol{\mu}_i$
——E-step2——
$\boldsymbol{U} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \, \boldsymbol{\mu}_i^T$

$\boldsymbol{S} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\mu}_i \, \boldsymbol{\mu}_i^T + \boldsymbol{\Sigma}$
——M-step——
$\eta_W = \eta_\Psi = \eta$
$\boldsymbol{E} = \boldsymbol{C} - \boldsymbol{U} \, \boldsymbol{W}^T - \boldsymbol{W} \, \boldsymbol{U} + \boldsymbol{W} \, \boldsymbol{S} \, \boldsymbol{W}^T$
——$\boldsymbol{W}$ update——
$\boldsymbol{W} = \boldsymbol{W} + \eta_W \left(\boldsymbol{U} \, \boldsymbol{S}^{-1} - \boldsymbol{W}\right)$
——diagonal $\boldsymbol{\Psi}$ update——
for all $1 \leq k \leq m$ **do**
$\Psi_{kk} = \Psi_{kk} + \eta_\Psi \, (E_{kk} - \Psi_{kk})$
end for
——full $\boldsymbol{\Psi}$ update——
$\boldsymbol{\Psi} = \boldsymbol{\Psi} + \eta_\Psi \, (\boldsymbol{E} - \boldsymbol{\Psi})$
——bound parameters——
$\boldsymbol{W} = \text{median}\{-W_{\max} \, , \, \boldsymbol{W} \, , \, W_{\max}\}$
$\boldsymbol{\Psi} = \text{median}\{\Psi_{\min} \, , \, \boldsymbol{\Psi} \, , \, \max\{\boldsymbol{C}\}\}$
if stopping criterion is met: STOP=true
end while

Algorithm 2 Projection with E-Step Improvement

Goal
obtain $\boldsymbol{\mu}_i^{\text{new}} = \boldsymbol{\mu}_i$ that decrease the E-step objective
Input
$\boldsymbol{\Sigma}^{\text{new}} = \boldsymbol{\Sigma}_p$, $\boldsymbol{\Sigma}^{\text{old}} = \boldsymbol{\Sigma}_p^{\text{old}}$
for $1 \leq i \leq n$: $(\boldsymbol{\mu}_p)_i$, $\boldsymbol{\mu}_i^{\text{old}}$, $p_i = \mathcal{N}((\boldsymbol{\mu}_p)_i, \boldsymbol{\Sigma}_p)$
simple projection P (rectified or rectified & normalized),
E-step objective: $O = \frac{1}{n} \sum_{i=1}^{n} D_{\text{KL}}(Q_i \parallel p_i)$
$\gamma_{\min}$, $\lambda_{\min}$, ρ_γ, ρ_λ, ϵ (for ϵ-active set)
Main
——Simple Projection——
perform `Newton Projection` by Algorithm 4 or Algorithm 3
——Scaled Projection——
if $0 \leq \Delta O$ **then**
following loop for: (1) γ, (2) λ, or (3) γ and λ annealing
$\gamma = \lambda = 1$
while $0 \leq \Delta O$ and $\lambda > \lambda_{\min}$ and $\gamma > \gamma_{\min}$ **do**
$\gamma = \rho_\gamma \gamma$ (skipped for λ annealing)
$\lambda = \rho_\lambda \lambda$ (skipped for γ annealing)
perform `Scaled Newton Projection` by Algorithm 5
end while
end if
——Scaled Projection With Reduced Matrix——
if $0 \leq \Delta O$ **then**
determine ϵ-active set as all j with $\mu_j \leq \epsilon$
set $\boldsymbol{H}$ to $\boldsymbol{\Sigma}_p^{-1}$ with ϵ-active columns and rows j fixed to $\boldsymbol{e}_j$
following loop for: (1) γ, (2) λ, or (3) γ and λ annealing
$\gamma = \lambda = 1$
while $0 \leq \Delta O$ and $\lambda > \lambda_{\min}$ and $\gamma > \gamma_{\min}$ **do**
$\gamma = \rho_\gamma \gamma$ (skipped for λ annealing)
$\lambda = \rho_\lambda \lambda$ (skipped for γ annealing)
perform `Scaled Projection With Reduced Matrix` by Algorithm 6
end while
end if
——General Gradient Projection——
while $0 \leq \Delta O$ **do**
use generalized reduced gradient Abadie and Carpentier [1969] OR
use Rosen's gradient projection Rosen [1961] OR
use method of Haug and Arora Haug and Arora [1979]
end while

Algorithm 3 Simple Projection: Rectifying

Goal
for $1 \leq i \leq n$: project $(\boldsymbol{\mu}_p)_i$ onto feasible set giving $\boldsymbol{\mu}_i$
Input
$(\boldsymbol{\mu}_p)_i$
Main
for all $1 \leq j \leq l$ **do**
$\mu_{ij} = \max\left\{0, [(\boldsymbol{\mu}_p)_i]_j\right\}$
end for

Algorithm 4 Simple Projection: Rectifying and Normalization

Goal
for $1 \leq i \leq n$: project $(\boldsymbol{\mu}_p)_i$ onto feasible set giving $\boldsymbol{\mu}_i$
Input
for $1 \leq i \leq n$: $(\boldsymbol{\mu}_p)_i$
Rectifier
for all $1 \leq i \leq n$ **do**
for all $1 \leq j \leq l$ **do**
$\hat{\mu}_{ij} \;=\; \max\left\{0, [(\boldsymbol{\mu}_p)_i]_j\right\}$
end for
end for
Normalizer
for all $1 \leq i \leq n$ **do**
if at least one $\hat{\mu}_{ij} > 0$ **then**
for all $1 \leq j \leq l$ **do**
$\mu_{ij} \;=\; \frac{\hat{\mu}_{ij}}{\sqrt{\frac{1}{n}\;\sum_{s=1}^{n}\hat{\mu}_{sj}^2}}$
end for
else
for all $1 \leq j \leq l$ **do**
$\mu_{ij} \;=\; \begin{cases} \sqrt{n} & \text{for} \quad j = \arg\max_{\hat{j}}\{[(\boldsymbol{\mu}_p)_i]_{\hat{j}}\} \\ 0 & \text{otherwise} \end{cases}$
end for
end if
end for

Algorithm 5 Scaled Newton Projection

Goal
perform a scaled Newton step with subsequent projection
Input
for $1 \leq i \leq n$: $(\boldsymbol{\mu}_p)_i$
for $1 \leq i \leq n$: $\boldsymbol{\mu}_i^{\text{old}}$
simple projection P (rectified or rectified & normalized),
λ (gradient step size), γ (projection difference)
Main
$\boldsymbol{d} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_i^{\text{old}} \;+\; \lambda\,((\boldsymbol{\mu}_p)_i \;-\; \boldsymbol{\mu}_i^{\text{old}})\right)$
$\boldsymbol{\mu}_i^{\text{new}} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_i^{\text{old}} \;+\; \gamma\left(\boldsymbol{d} \;-\; \boldsymbol{\mu}_i^{\text{old}}\right)\right)$

Algorithm 6 Scaled Projection With Reduced Matrix

Goal
perform a scaled projection step with reduced matrix
Input
for $1 \leq i \leq n$: $(\boldsymbol{\mu}_p)_i$
for $1 \leq i \leq n$: $\boldsymbol{\mu}_i^{\text{old}}$
simple projection P (rectified or rectified & normalized),
λ, γ, $\boldsymbol{H}$, $\boldsymbol{\Sigma}_p^{-1}$
Main
$\boldsymbol{d} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_i^{\text{old}} \;+\; \lambda\,\boldsymbol{H}^{-1}\,\boldsymbol{\Sigma}_p^{-1}((\boldsymbol{\mu}_p)_i \;-\; \boldsymbol{\mu}_i^{\text{old}})\right)$
$\boldsymbol{\mu}_i^{\text{new}} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_i^{\text{old}} \;+\; \gamma\left(\boldsymbol{d} \;-\; \boldsymbol{\mu}_i^{\text{old}}\right)\right)$

Algorithm 7 Weight Decay

Input
Parameters $\boldsymbol{W}$
Weight decay factors γ_G (Gaussian) and γ_L (Laplacian)
Gaussian
$\boldsymbol{W} \ = \ \boldsymbol{W} \ - \ \gamma_G \, \boldsymbol{W}$
Laplacian
$\hat{\boldsymbol{W}} \ = \ \text{median}\{-\gamma_L \ , \ \boldsymbol{W} \ , \ \gamma_L\}$
$\boldsymbol{W} \ = \ \boldsymbol{W} \ - \ \hat{\boldsymbol{W}}$

Algorithm 8 Dropout

Input
for $1 \leq i \leq n$: $\boldsymbol{\mu}_i$
dropout probability d
Main
for all $1 \leq i \leq n$ **do**
for all $1 \leq j \leq l$ **do**
$\Pr(\delta = 0) \ = \ d$
$\mu_{ij} \ = \ \delta \, \mu_{ij}$
end for
end for

3 Convergence Proof for the RFN Learning Algorithm

Theorem 1 (RFN Convergence)**.** *The rectified factor network (RFN) learning algorithm given in Algorithm 1 is a "generalized alternating minimization" (GAM) algorithm and converges to a solution that maximizes the objective $\mathcal{F}$.*

Proof. The factor analysis EM algorithm is given by Eq. (67) and Eq. (68) in Section 5. Algorithm 1 is the factor analysis EM algorithm with modified the E-step and the M-step. The E-step is modified by constraining the variational distribution Q to non-negative means and by normalizing its means across the samples. The M-step is modified to a Newton direction gradient step.

Like EM factor analysis, Algorithm 1 aims at maximizing the negative *free energy* $\mathcal{F}$, which is

$$
\begin{aligned}
\mathcal{F} \ &= \ \frac{1}{n} \ \sum_{i=1}^{n} \log p(\boldsymbol{v}_i) \ - \ \frac{1}{n} \ \sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)) \qquad (1) \\
&= \ \frac{1}{n} \ \sum_{i=1}^{n} \int Q(\boldsymbol{h}_i) \ \log p(\boldsymbol{v}_i) \ d\boldsymbol{h}_i \ - \ \frac{1}{n} \ \sum_{i=1}^{n} \int Q(\boldsymbol{h}_i) \ \log \frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)} \ d\boldsymbol{h}_i \\
&= \ - \ \frac{1}{n} \ \sum_{i=1}^{n} \int Q(\boldsymbol{h}_i) \ \log \frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i, \boldsymbol{v}_i)} \ d\boldsymbol{h}_i \\
&= \ - \ \frac{1}{n} \ \sum_{i=1}^{n} \int Q(\boldsymbol{h}_i) \ \log \frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i)} \ d\boldsymbol{h}_i \ + \ \frac{1}{n} \ \sum_{i=1}^{n} \int Q(\boldsymbol{h}_i) \ \log p(\boldsymbol{v}_i \mid \boldsymbol{h}_i) \ d\boldsymbol{h}_i \\
&= \ \frac{1}{n} \ \sum_{i=1}^{n} \int Q(\boldsymbol{h}_i) \ \log p(\boldsymbol{v}_i \mid \boldsymbol{h}_i) \ d\boldsymbol{h}_i \ - \frac{1}{n} \ \sum_{i=1}^{n} \ D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i)) \ .
\end{aligned}
$$

D_{KL} denotes the Kullback-Leibler (KL) divergence Kullback and Leibler [1951], which is larger than or equal to zero.

Algorithm 1 decreases $\frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i))$ (the E-step objective) in its E-step under constraints for non-negative means and normalization. The constraint optimization problem from

Section 9.2 for the E-step is

$$\begin{aligned} \min_{Q(\boldsymbol{h}_i)} \quad & \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)) \\ \text{s.t.} \quad & \forall_i : \boldsymbol{\mu}_i \geq \mathbf{0} \,, \\ & \forall_j : \frac{1}{n} \sum_{i=1}^{n} \mu_{ij}^2 = 1 \,. \end{aligned} \tag{2}$$

The M-step of Algorithm 1 aims at decreasing

$$\mathcal{E} = -\frac{1}{n} \sum_{i=1}^{n} \int_{\mathbb{R}^l} Q(\boldsymbol{h}_i) \, \log\left(p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\right) \, d\boldsymbol{h}_i \,. \tag{3}$$

Algorithm 1 performs one gradient descent step in the Newton direction to decrease $\mathcal{E}$, while EM factor analysis minimizes $\mathcal{E}$.

From the modification of the E-step and the M-step follows that Algorithm 1 is a *Generalized Alternating Minimization (GAM)* algorithm according to Gunawardana and Byrne [2005]. GAM is an EM algorithm that increases $\mathcal{F}$ in the E-step and increases $\mathcal{F}$ in the M-step (see also Section 7). The most important requirements for the convergence of the GAM algorithm according to Theorem 4 (Proposition 5 in Gunawardana and Byrne [2005]) are the increase of the objective $\mathcal{F}$ in both the E-step and the M-step. Therefore we first show these two decreases before showing that all requirements of convergence Theorem 4 are met.

Algorithm 1 ensures to decrease the M-step objective. The M-step objective $\mathcal{E}$ is convex in $\boldsymbol{W}$ and $\boldsymbol{\Psi}^{-1}$ according to Theorem 5 and Theorem 7. The update with $\eta_W = \eta_\Psi = \eta = 1$ leads to the minimum of $\mathcal{E}$ according to Theorem 5 and Theorem 7. The convexity of $\mathcal{E}$ guarantees that each update with $0 < \eta_W = \eta_\Psi = \eta \leq 1$ decreases the M-step objective $\mathcal{E}$, except the current $\boldsymbol{W}$ and $\boldsymbol{\Psi}^{-1}$ are already the minimizers.

Algorithm 1 ensures to decrease the E-step objective. The E-step decrease of Algorithm 1 is performed by Algorithm 2. According to Theorem 11 the scaled projection with reduced matrix ensures a decrease of the E-step objective for rectifying constraints (convex feasible set). According to Theorem 10 also gradient projection methods ensure a decrease of the E-step objective for rectifying constraints. For rectifying constraints and normalization, the feasible set is not convex because of the equality constraints. To optimize such problems, the generalized reduced gradient method Abadie and Carpentier [1969] solves each equality constraint for one variable and inserts it into the objective. For our problem Eq. (146) gives the solution and Eq. (147) the resulting convex constraints. Now scaled projection and gradient projection methods can be applied. For rectifying and normalizing constraints, also Rosen's Rosen [1961] and Haug & Arora's Haug and Arora [1979] gradient projection method ensures a decrease of the E-step objective since they can be applied to non-convex problems.

We show that the requirements as given in Section 7 for GAM convergence according to Theorem 4 (Proposition 5 in Gunawardana and Byrne [2005]) are fulfilled:

1. the learning rules, that is, the E-step and the M-step, are closed maps $\longrightarrow$ ensured by continuous and continuous differentiable maps,

2. the parameter set is compact $\longrightarrow$ ensured by bounding $\boldsymbol{\Psi}$ and $\boldsymbol{W}$,

3. the family of variational distributions is compact (often described by the feasible set of parameters of the variational distributions) $\longrightarrow$ ensured by continuous and continuous differentiable functions for the constraints and by the bounds on the variational parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ determined by bounds on the parameters and the data,

4. the support of the density models does not depend on the parameter $\longrightarrow$ ensured by Gaussian models with full-rank covariance matrix,

5. the density models are continuous in the parameters $\longrightarrow$ ensured by Gaussian models

6. the E-step has a unique maximizer $\longrightarrow$ ensured by the convex, continuous, and continuous differentiable function that is minimized Dredze et al. [2008, 2012] together with compact feasible set for the variational parameters, the maximum may be local for non-convex feasible sets stemming from normalization,

7. the E-step increases the objective if not at the maximizer $\longrightarrow$ ensured as shown above,

8. the M-step has a unique maximizer (this is not required) $\longrightarrow$ ensured by minimizing a convex, continuous and continuous differentiable function in the model parameter and a convex feasible set, the maximum is a global maximum,

9. the M-step increases the objective if not at the maximizer $\longrightarrow$ ensured as shown above.

□

Since this Proposition 5 in Gunawardana and Byrne [2005] is based on Zangwill's generalized convergence theorem, updates of the RFN algorithm are viewed as point-to-set mappings Zangwill [1969]. Therefore the numerical precision, the choice of the methods in the E-step, and GPU implementations are covered by the proof. That the M-step has a unique maximizer is not required to proof Theorem 1 by Theorem 4. However we obtain an alternative proof by exchanging the variational distribution Q and the parameters $(\boldsymbol{W}, \boldsymbol{\Psi})$, that is, exchanging the E-step and the M-step. A theorem analog to Theorem 4 but with E-step and M-step conditions exchanged can be derived from Zangwill's generalized convergence theorem Zangwill [1969].

The resulting model from the GAM procedure is at a local maximum of the objective given the model family and the family of variational distributions. *The solution minimizes the KL-distance between the family of full variational distributions and full model family*. "Full" means that both the observed and the hidden variables are taken into account, where for the variational distributions the probability of the observations is set to 1. The *desired family* is defined as the set of all probability distributions that assign probability one to the observation. In our case the family of variational distributions is not the desired family since some distributions are excluded by the constraints. Therefore the solution of the GAM optimization does not guarantee stationary points in likelihood Gunawardana and Byrne [2005]. This means that we do not maximize the likelihood but minimize

$$-\mathcal{F} \;\approx\; D_{\mathrm{KL}}(Q(\boldsymbol{h}, \boldsymbol{v}) \parallel p(\boldsymbol{h}, \boldsymbol{v})) \;+\; c \tag{4}$$

according to Eq. (73), where c is a constant independent of Q and independent of the model parameters.

4 Correctness Proofs for the RFN Learning Algorithms

The RFN algorithm is correct if it has a low reconstruction error and explains the data covariance matrix by its parameters like factor analysis. We show in Theorem 2 and Theorem 3 that the RFN algorithm

1. minimizes the reconstruction error given $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}$ (the error is quadratic in $\boldsymbol{\Psi}$);
2. explains the covariance matrix by its parameters $\boldsymbol{W}$ and $\boldsymbol{\Psi}$ plus an estimate of the second moment of the coding units $\boldsymbol{S}$.

Since the minimization of the reconstruction error is based on $\boldsymbol{\mu}_i$, the quality of reconstruction and covariance explanation depends on the correlation between $\boldsymbol{\mu}_i$ and $\boldsymbol{v}_i$. The larger the correlation between $\boldsymbol{\mu}_i$ and $\boldsymbol{v}_i$, the lower the reconstruction error and the better the explanation of the data covariance. We ensure maximal information in $\boldsymbol{\mu}_i$ on $\boldsymbol{v}_i$ by the I-projection (the minimal Kullback-Leibler distance) of the posterior onto the family of rectified and normalized Gaussian distributions.

The reconstruction error for given mean values $\boldsymbol{\mu}_i$ is

$$\frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{\epsilon}_i\|_2^2 \;, \tag{5}$$

where

$$\boldsymbol{\epsilon}_i \;=\; \boldsymbol{v}_i \;-\; \boldsymbol{W}\;\boldsymbol{\mu}_i\;. \tag{6}$$

The reconstruction error for using the whole variational distribution $Q(\boldsymbol{h}_i)$ instead of its means is $\boldsymbol{\Psi}$. Below we will derive Eq. (17), which is

$$\boldsymbol{\Psi} \;=\; \mathrm{diag}\left(\frac{1}{n}\;\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\;\boldsymbol{\epsilon}_i^T \;+\; \boldsymbol{W}\;\boldsymbol{\Sigma}\;\boldsymbol{W}^T\right)\;. \tag{7}$$

Therefore $\boldsymbol{\Psi}$ is the reconstruction error for given mean values plus the variance $\boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T$ introduced by the hidden variables.

4.1 Diagonal Noise Covariance Update

Theorem 2 (RFN Correctness: Diagonal Noise Covariance Update)**.** *The fixed point $\boldsymbol{W}$ minimizes* $\mathrm{Tr}\left(\boldsymbol{\Psi}\right)$ *given $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}$ by ridge regression with*

$$\mathrm{Tr}\left(\boldsymbol{\Psi}\right) \;=\; \frac{1}{n}\;\sum_{i=1}^{n}\left\|\boldsymbol{\epsilon}_i\right\|_2^2 \;+\; \left\|\boldsymbol{W}\;\boldsymbol{\Sigma}^{1/2}\right\|_{\mathrm{F}}^2\;, \tag{8}$$

where we used the error

$$\boldsymbol{\epsilon}_i \;=\; \boldsymbol{v}_i \;-\; \boldsymbol{W}\;\boldsymbol{\mu}_i \tag{9}$$

The model explains the data covariance matrix by

$$\boldsymbol{C} \;=\; \boldsymbol{\Psi} \;+\; \boldsymbol{W}\;\boldsymbol{S}\;\boldsymbol{W}^T \tag{10}$$

up to an error, which is quadratic in $\boldsymbol{\Psi}$ for $\boldsymbol{\Psi} \ll \boldsymbol{W}\boldsymbol{W}^T$. The reconstruction error

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{\epsilon}_i\right\|_2^2 \tag{11}$$

is quadratic in $\boldsymbol{\Psi}$ for $\boldsymbol{\Psi} \ll \boldsymbol{W}\boldsymbol{W}^T$.

Proof. The fixed point equation for the $\boldsymbol{W}$ update is

$$\Delta\boldsymbol{W} \;=\; \boldsymbol{U}\;\boldsymbol{S}^{-1} \;-\; \boldsymbol{W} \;=\; \boldsymbol{0} \;\Rightarrow\; \boldsymbol{W} \;=\; \boldsymbol{U}\;\boldsymbol{S}^{-1}\;. \tag{12}$$

Using the definition of $\boldsymbol{U}$ and $\boldsymbol{S}$, the fixed point equation Eq. (12) gives

$$\boldsymbol{W} \;=\; \left(\frac{1}{n}\;\sum_{i=1}^{n}\boldsymbol{v}_i\;\boldsymbol{\mu}_i^T\right)\;\left(\frac{1}{n}\;\sum_{i=1}^{n}\boldsymbol{\mu}_i\;\boldsymbol{\mu}_i^T \;+\; \boldsymbol{\Sigma}\right)^{-1} \tag{13}$$

Therefore $\boldsymbol{W}$ is a *ridge regression* estimate, also called *generalized Tikhonov regularization* estimate, which minimizes

$$\begin{aligned}
&\frac{1}{n}\;\sum_{i=1}^{n}\left\|\boldsymbol{v}_i \;-\; \boldsymbol{W}\;\boldsymbol{\mu}_i\right\|_2^2 \;+\; \left\|\boldsymbol{W}\;\boldsymbol{\Sigma}^{1/2}\right\|_{\mathrm{F}}^2 \qquad (14)\\
&=\; \frac{1}{n}\;\sum_{i=1}^{n}\left\|\boldsymbol{\epsilon}_i\right\|_2^2 \;+\; \left\|\boldsymbol{W}\;\boldsymbol{\Sigma}^{1/2}\right\|_{\mathrm{F}}^2\\
&=\; \frac{1}{n}\;\sum_{i=1}^{n}\boldsymbol{\epsilon}_i^T\;\boldsymbol{\epsilon}_i \;+\; \mathrm{Tr}\left(\boldsymbol{W}\;\boldsymbol{\Sigma}^{1/2}\;\boldsymbol{\Sigma}^{1/2}\boldsymbol{W}^T\right)\\
&=\; \mathrm{Tr}\left(\frac{1}{n}\;\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\;\boldsymbol{\epsilon}_i^T \;+\; \boldsymbol{W}\;\boldsymbol{\Sigma}\;\boldsymbol{W}^T\right)\;,
\end{aligned}$$

where we used the reconstruction error

$$\boldsymbol{\epsilon}_i \;=\; \boldsymbol{v}_i \;-\; \boldsymbol{W}\;\boldsymbol{\mu}_i\;. \tag{15}$$

We obtain with this definition of the error

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\,\boldsymbol{\epsilon}_i^T \;+\; \boldsymbol{W}\,\boldsymbol{\Sigma}\,\boldsymbol{W}^T \qquad (16)\\
&= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\,\boldsymbol{v}_i^T \;-\; \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\,\boldsymbol{\mu}_i^T\,\boldsymbol{W}^T \;-\; \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{W}\,\boldsymbol{\mu}_i\,\boldsymbol{v}_i^T\\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{W}\,\boldsymbol{\mu}_i\,\boldsymbol{\mu}_i^T\,\boldsymbol{W}^T \;+\; \boldsymbol{W}\,\boldsymbol{\Sigma}\,\boldsymbol{W}^T\\
&= \boldsymbol{C} \;-\; \boldsymbol{U}\,\boldsymbol{W}^T \;-\; \boldsymbol{W}\,\boldsymbol{U}^T \;+\; \boldsymbol{W}\,\boldsymbol{S}\,\boldsymbol{W}^T\,.
\end{aligned}$$

Therefore from the fixed point equation for $\boldsymbol{\Psi}$ with the diagonal update rule follows

$$\boldsymbol{\Psi} \;=\; \mathrm{diag}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\,\boldsymbol{\epsilon}_i^T \;+\; \boldsymbol{W}\,\boldsymbol{\Sigma}\,\boldsymbol{W}^T\right)\,, \qquad (17)$$

where "diag" projects a matrix to a diagonal matrix. From this follows that

$$\mathrm{Tr}\,(\boldsymbol{\Psi}) \;=\; \mathrm{Tr}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\,\boldsymbol{\epsilon}_i^T \;+\; \boldsymbol{W}\,\boldsymbol{\Sigma}\,\boldsymbol{W}^T\right)\,. \qquad (18)$$

Consequently, the fixed point $\boldsymbol{W}$ minimizes $\mathrm{Tr}\,(\boldsymbol{\Psi})$ given $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}$.

After convergence of the algorithm $\boldsymbol{\Sigma} \;=\; \left(\boldsymbol{I} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}$ holds. The Woodbury identity (matrix inversion lemma) states

$$\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \;=\; \boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1}\boldsymbol{W}\left(\boldsymbol{I} \;+\; \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\boldsymbol{W}^T\boldsymbol{\Psi}^{-1} \qquad (19)$$

from which follows by multiplying the equation from right and left by $\boldsymbol{\Psi}$ that

$$\begin{aligned}
\boldsymbol{W}\,\boldsymbol{\Sigma}\,\boldsymbol{W}^T \;&=\; \boldsymbol{W}\,\left(\boldsymbol{I} \;+\; \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,\boldsymbol{W}^T \qquad (20)\\
&=\; \boldsymbol{\Psi} \;-\; \boldsymbol{\Psi}\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\,\boldsymbol{\Psi}
\end{aligned}$$

Inserting this equation Eq. (20) into Eq. (17) gives

$$\begin{aligned}
\boldsymbol{\Psi} \;&=\; \mathrm{diag}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\,\boldsymbol{\epsilon}_i^T \;+\; \boldsymbol{\Psi} \;-\; \boldsymbol{\Psi}\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\,\boldsymbol{\Psi}\right) \qquad (21)\\
&=\; \boldsymbol{\Psi} \;+\; \mathrm{diag}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\,\boldsymbol{\epsilon}_i^T \;-\; \boldsymbol{\Psi}\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\,\boldsymbol{\Psi}\right)\,.
\end{aligned}$$

Therefore we have

$$\mathrm{diag}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\,\boldsymbol{\epsilon}_i^T \;-\; \boldsymbol{\Psi}\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\,\boldsymbol{\Psi}\right) \;=\; \boldsymbol{0}\,. \qquad (22)$$

It follows that

$$\begin{aligned}
\mathrm{Tr}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\,\boldsymbol{\epsilon}_i^T\right) \;&=\; \mathrm{Tr}\left(\boldsymbol{\Psi}\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\,\boldsymbol{\Psi}\right)\\
&\leq\; \mathrm{Tr}\left(\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\right)\,\mathrm{Tr}\,(\boldsymbol{\Psi})^2\,. \qquad (23)
\end{aligned}$$

The inequality uses the fact that for positive definite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ inequality $\mathrm{Tr}(\boldsymbol{AB}) \leq \mathrm{Tr}(\boldsymbol{A})\mathrm{Tr}(\boldsymbol{B})$ holds Patel and Toda [1979]. Thus, for $\boldsymbol{\Psi} \ll \boldsymbol{W}\boldsymbol{W}^T$ the error $\mathrm{Tr}\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^T\right) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\epsilon}_i^T\boldsymbol{\epsilon}_i$ is quadratic in $\boldsymbol{\Psi}$.

Multiplying the fixed point equation Eq. (12) by $\boldsymbol{S}$ gives $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{S}$. Therefore we have:

$$\boldsymbol{W}\,\boldsymbol{U}^T \;=\; \boldsymbol{W}\,\boldsymbol{S}\,\boldsymbol{W}^T \;=\; \boldsymbol{U}\,\boldsymbol{W}^T\,. \qquad (24)$$

Inserting Eq. (20) into the first line of Eq. (16) and Eq. (24) for simplifying the last line of Eq. (16) gives

$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\epsilon}_i \, \boldsymbol{\epsilon}_i^T \; - \; \boldsymbol{\Psi} \, \left(\boldsymbol{W} \, \boldsymbol{W}^T \; + \; \boldsymbol{\Psi} \right)^{-1} \, \boldsymbol{\Psi} \; = \; \boldsymbol{C} \; - \; \boldsymbol{\Psi} \; - \; \boldsymbol{W} \, \boldsymbol{S} \, \boldsymbol{W}^T \, . \tag{25}$$

Using the trace norm (nuclear norm or Ky-Fan n-norm) on matrices, Eq. (23) states that the left hand side is quadratic in $\boldsymbol{\Psi}$ for $\boldsymbol{\Psi} \ll \boldsymbol{W}\boldsymbol{W}^T$. The trace norm of a positive semi-definite matrix is its trace and bounds the Frobenius norm Srebro [2004]. Furthermore, Eq. (22) states that the left hand side of this equation has zero diagonal entries. Therfore it follows that

$$\boldsymbol{C} \; = \; \boldsymbol{\Psi} \; + \; \boldsymbol{W} \, \boldsymbol{S} \, \boldsymbol{W}^T \tag{26}$$

holds except an error, which is quadratic in $\boldsymbol{\Psi}$ for $\boldsymbol{\Psi} \ll \boldsymbol{W}\boldsymbol{W}^T$. The diagonal is exactly modeled according to Eq. (22). □

Therefore the model corresponding to the fixed point explains the empirical matrix of second moments $\boldsymbol{C}$ by a noise part $\boldsymbol{\Psi}$ and a signal part $\boldsymbol{W}\boldsymbol{S}\boldsymbol{W}^T$. Like factor analysis the data variance is explained by the model via the parameters $\boldsymbol{\Psi}$ (noise) and $\boldsymbol{W}$ (signal).

4.2 Full Noise Covariance Update

Theorem 3 (RFN Correctness: Full Noise Covariance Update)**.** *The fixed point $\boldsymbol{W}$ minimizes* $\mathrm{Tr}\left(\boldsymbol{\Psi}\right)$ *given $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}$ by ridge regression with*

$$\mathrm{Tr}\left(\boldsymbol{\Psi}\right) \; = \; \frac{1}{n} \sum_{i=1}^{n} \left\| \boldsymbol{\epsilon}_i \right\|_2^2 \; + \; \left\| \boldsymbol{W} \, \boldsymbol{\Sigma}^{1/2} \right\|_{\mathrm{F}}^2 \, , \tag{27}$$

where we used the error

$$\boldsymbol{\epsilon}_i \; = \; \boldsymbol{v}_i \; - \; \boldsymbol{W} \, \boldsymbol{\mu}_i \tag{28}$$

The model explains the data covariance matrix by

$$\boldsymbol{C} \; = \; \boldsymbol{\Psi} \; + \; \boldsymbol{W} \, \boldsymbol{S} \, \boldsymbol{W}^T \, . \tag{29}$$

The reconstruction error

$$\frac{1}{n} \sum_{i=1}^{n} \left\| \boldsymbol{\epsilon}_i \right\|_2^2 \tag{30}$$

is quadratic in $\boldsymbol{\Psi}$ for $\boldsymbol{\Psi} \ll \boldsymbol{W}\boldsymbol{W}^T$.

Proof. The first part follows from previous Theorem 2. The fixed point equation for the $\boldsymbol{\Psi}$ update is

$$\boldsymbol{\Psi} \; = \; \boldsymbol{C} \; - \; \boldsymbol{U} \, \boldsymbol{W}^T \; - \; \boldsymbol{W} \, \boldsymbol{U}^T \; + \; \boldsymbol{W} \, \boldsymbol{S} \, \boldsymbol{W}^T \, , \tag{31}$$

using Eq. (24) this leads to

$$\boldsymbol{C} \; = \; \boldsymbol{\Psi} \; + \; \boldsymbol{W} \, \boldsymbol{S} \, \boldsymbol{W}^T \, . \tag{32}$$

From Eq. (16) follows for the fixed point of $\boldsymbol{\Psi}$ with the full update rule:

$$\boldsymbol{\Psi} \; = \; \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\epsilon}_i \, \boldsymbol{\epsilon}_i^T \; + \; \boldsymbol{W} \, \boldsymbol{\Sigma} \, \boldsymbol{W}^T \, . \tag{33}$$

Inserting Eq. (20) into Eq. (33) gives

$$\boldsymbol{\Psi} \; = \; \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\epsilon}_i \, \boldsymbol{\epsilon}_i^T \; + \; \boldsymbol{\Psi} \; - \; \boldsymbol{\Psi} \, \left(\boldsymbol{W} \, \boldsymbol{W}^T \; + \; \boldsymbol{\Psi} \right)^{-1} \, \boldsymbol{\Psi} \, , \tag{34}$$

from which follows

$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\epsilon}_i \, \boldsymbol{\epsilon}_i^T \; = \; \boldsymbol{\Psi} \, \left(\boldsymbol{W} \, \boldsymbol{W}^T \; + \; \boldsymbol{\Psi} \right)^{-1} \, \boldsymbol{\Psi} \, . \tag{35}$$

Thus, the error $\mathrm{Tr}\left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T \right) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\epsilon}_i^T \boldsymbol{\epsilon}_i$ is quadratic in $\boldsymbol{\Psi}$, for $\boldsymbol{\Psi} \ll \boldsymbol{W}\boldsymbol{W}^T$. □

5 Maximum Likelihood Factor Analysis

We are given the data $\{\boldsymbol{v}\} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ which is assumed to be centered. Centering can be done by subtracting the mean $\boldsymbol{\mu}$ from the data. The model is

$$\boldsymbol{v} \;=\; \boldsymbol{W}\boldsymbol{h} \;+\; \boldsymbol{\epsilon}\,, \tag{36}$$

where

$$\boldsymbol{h} \;\sim\; \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}\right) \quad \text{and} \quad \boldsymbol{\epsilon} \;\sim\; \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Psi}\right)\,. \tag{37}$$

The model includes the *observations* $\boldsymbol{v} \in \mathbb{R}^m$, the *noise* $\boldsymbol{\epsilon} \in \mathbb{R}^m$, the *factors* $\boldsymbol{h} \in \mathbb{R}^l$, the *factor loading matrix* $\boldsymbol{W} \in \mathbb{R}^{m \times l}$, and the *noise covariance matrix* $\boldsymbol{\Psi} \in \mathbb{R}^{m \times m}$. Typically we assume that $\boldsymbol{\Psi}$ is a diagonal matrix to explain data covariance by signal and not by noise. The data variance is explained through a signal part $\boldsymbol{W}\boldsymbol{h}$ and through a noise part $\boldsymbol{\epsilon}$. The parameters of the model are $\boldsymbol{W}$ and $\boldsymbol{\Psi}$. From the model assumption it follows that if $\boldsymbol{h}$ is given, then only the noise $\boldsymbol{\epsilon}$ is a random variable and we have

$$\boldsymbol{v} \mid \boldsymbol{h} \;\sim\; \mathcal{N}\left(\boldsymbol{W}\boldsymbol{h}, \boldsymbol{\Psi}\right)\,. \tag{38}$$

We want to derive the *likelihood* of the data under the model, that is, the likelihood that the model has produced the data. Let E denote the expectation of the data including the prior distribution of the factors and the noise distribution. We obtain for the first two moments and the variance:

$$\begin{aligned}
\mathrm{E}(\boldsymbol{v}) \;&=\; \mathrm{E}(\boldsymbol{W}\boldsymbol{h} \;+\; \boldsymbol{\epsilon}) \;=\; \boldsymbol{W}\mathrm{E}(\boldsymbol{h}) \;+\; \mathrm{E}(\boldsymbol{\epsilon}) \;=\; \boldsymbol{0}\,, \qquad (39)\\
\mathrm{E}\left(\boldsymbol{v}\,\boldsymbol{v}^T\right) \;&=\; \mathrm{E}\left((\boldsymbol{W}\boldsymbol{h} \;+\; \boldsymbol{\epsilon})(\boldsymbol{W}\boldsymbol{h} \;+\; \boldsymbol{\epsilon})^T\right) \;=\\
&\quad \boldsymbol{W}\mathrm{E}\left(\boldsymbol{h}\,\boldsymbol{h}^T\right)\boldsymbol{W}^T \;+\; \boldsymbol{W}\mathrm{E}\left(\boldsymbol{h}\right)\mathrm{E}\left(\boldsymbol{\epsilon}^T\right)\\
&\quad +\; \mathrm{E}\left(\boldsymbol{\epsilon}\right)\,\mathrm{E}\left(\boldsymbol{h}^T\right)\,\boldsymbol{W}^T \;+\; \mathrm{E}\left(\boldsymbol{\epsilon}\,\boldsymbol{\epsilon}^T\right) \;=\\
&\quad \boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\\
\mathrm{var}(\boldsymbol{v}) \;&=\; \mathrm{E}\left(\boldsymbol{v}\,\boldsymbol{v}^T\right) \;-\; \left(\mathrm{E}(\boldsymbol{v})\right)^2 \;=\; \boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\,. \qquad (40)
\end{aligned}$$

The observations are Gaussian distributed since their distribution is the product of two Gaussian densities divided by a normalizing constant. Therefore, the marginal distribution for $\boldsymbol{v}$ is

$$\boldsymbol{v} \;\sim\; \mathcal{N}\left(\boldsymbol{0}\,,\; \boldsymbol{W}\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)\,. \tag{41}$$

The log-likelihood $\log \prod_{i=1}^n p(\boldsymbol{v}_i)$ of the data $\{\boldsymbol{v}\}$ under the model $(\boldsymbol{W}, \boldsymbol{\Psi})$ is

$$\begin{aligned}
&\log \prod_{i=1}^n p(\boldsymbol{v}_i) \;=\; \log \prod_{i=1}^n (2\pi)^{-m/2} \left|\boldsymbol{W}\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right|^{-1/2} \qquad (42)\\
&\exp\left(-\frac{1}{2}\left(\boldsymbol{v}_i^T\left(\boldsymbol{W}\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\boldsymbol{v}_i\right)\right)\\
&= \;-\;\frac{n\,m}{2}\;\log\left(2\pi\right) - \;\frac{n}{2}\;\log\left|\boldsymbol{W}\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right|\\
&-\frac{1}{2}\sum_{i=1}^n \boldsymbol{v}_i^T\left(\boldsymbol{W}\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\boldsymbol{v}_i\,,
\end{aligned}$$

where $|.|$ denotes the absolute value of the determinant of a matrix.

To maximize the likelihood is difficult since a closed form for the maximum does not exists. Therefore, typically the expectation maximization (EM) algorithm is used to maximize the likelihood. For the EM algorithm a variational distribution Q is required which estimates the factors given the observations.

We consider a single data vector $\boldsymbol{v}_i$. The posterior is also Gaussian with mean $(\boldsymbol{\mu}_p)_i$ and covariance matrix $\boldsymbol{\Sigma}_p$:

$$\begin{aligned}
\boldsymbol{h}_i \mid \boldsymbol{v}_i \;&\sim\; \mathcal{N}\left((\boldsymbol{\mu}_p)_i, \boldsymbol{\Sigma}_p\right) \qquad (43)\\
(\boldsymbol{\mu}_p)_i \;&=\; \boldsymbol{W}^T\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\boldsymbol{v}_i\\
\boldsymbol{\Sigma}_p \;&=\; \boldsymbol{I} \;-\; \boldsymbol{W}^T\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W}\,,
\end{aligned}$$

where we used the fact that

$$\begin{aligned} \boldsymbol{a} \;\sim\; \mathcal{N}\left(\boldsymbol{\mu}_a, \Sigma_{aa}\right) \;,\; \boldsymbol{u} \;\sim\; \mathcal{N}\left(\boldsymbol{\mu}_u, \Sigma_{uu}\right) \;, \\ \Sigma_{ua} \;=\; \mathrm{Cov}(\boldsymbol{u}, \boldsymbol{a}) \text{ and } \;\Sigma_{au} \;=\; \mathrm{Cov}(\boldsymbol{a}, \boldsymbol{u}) : \\ \boldsymbol{a} \mid \boldsymbol{u} \;\sim\; \mathcal{N}\left(\boldsymbol{\mu}_a \;+\; \Sigma_{au} \Sigma_{uu}^{-1} \left(\boldsymbol{u} \;-\; \boldsymbol{\mu}_u\right) \;,\; \Sigma_{aa} \;-\; \Sigma_{au} \Sigma_{uu}^{-1} \Sigma_{ua}\right) \end{aligned} \tag{44}$$

and

$$\mathrm{E}(\boldsymbol{h}\boldsymbol{v}) \;=\; \boldsymbol{W}\; \mathrm{E}(\boldsymbol{h}\; \boldsymbol{h}^T) \;=\; \boldsymbol{W} \;. \tag{45}$$

The EM algorithm sets Q to the posterior distribution for data vector $\boldsymbol{v}_i$:

$$Q_i(\boldsymbol{h}_i) \;=\; p\left(\boldsymbol{h}_i \mid \boldsymbol{v}_i; \boldsymbol{W}, \boldsymbol{\Psi}\right) \;=\; \mathcal{N}\left((\boldsymbol{\mu}_p)_i, \boldsymbol{\Sigma}_p\right) \;, \tag{46}$$

therefore we obtain for standared EM

$$\boldsymbol{\mu}_i \;=\; (\boldsymbol{\mu}_q)_i \;=\; (\boldsymbol{\mu}_p)_i \tag{47}$$

$$\boldsymbol{\Sigma} \;=\; \boldsymbol{\Sigma}_q \;=\; \boldsymbol{\Sigma}_p \;. \tag{48}$$

The matrix inversion lemma (Woodbury identiy) can be used to compute $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}$:

$$\left(\boldsymbol{W}\; \boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \;=\; \boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1} \boldsymbol{W} \left(\boldsymbol{I} \;+\; \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \;. \tag{49}$$

Using this identity, the mean and the covariance matrix can be computed as:

$$\begin{aligned} \boldsymbol{\mu}_i \;&=\; \boldsymbol{W}^T \;\left(\boldsymbol{W}\; \boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \boldsymbol{v}_i \;=\; \left(\boldsymbol{I} \;+\; \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{v}_i \;, \\ \boldsymbol{\Sigma} \;&=\; \boldsymbol{I} \;-\; \boldsymbol{W}^T \;\left(\boldsymbol{W}\; \boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1} \boldsymbol{W} \;=\; \left(\boldsymbol{I} \;+\; \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \;. \end{aligned} \tag{50}$$

The EM algorithm maximizes a lower bound $\mathcal{F}$ on the log-likelihood:

$$\begin{aligned} \mathcal{F} \;&=\; \log p(\boldsymbol{v}_i) \;-\; D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)) \\ &=\; \int Q(\boldsymbol{h}_i) \;\log p(\boldsymbol{v}_i)\; d\boldsymbol{h}_i \;-\; \int Q(\boldsymbol{h}_i) \;\log \frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)}\; d\boldsymbol{h}_i \\ &=\; -\; \int Q(\boldsymbol{h}_i) \;\log \frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i, \boldsymbol{v}_i)}\; d\boldsymbol{h}_i \\ &=\; -\; \int Q(\boldsymbol{h}_i) \;\log \frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i)}\; d\boldsymbol{h}_i \;+\; \int Q(\boldsymbol{h}_i) \;\log p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\; d\boldsymbol{h}_i \\ &=\; \int Q(\boldsymbol{h}_i) \;\log p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\; d\boldsymbol{h}_i \;-\; D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i)) \;. \end{aligned} \tag{51}$$

D_{KL} denotes the Kullback-Leibler (KL) divergence Kullback and Leibler [1951] which is larger than zero.

$\mathcal{F}$ is the EM objective which has to be maximized in order to maximize the likelihood. The **E-step** maximizes $\mathcal{F}$ with respect to the variational distribution Q, therefore the E-step minimizes $D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i))$. After the standard unconstrained E-step, the variational distribution is equal to the posterior, i.e. $Q(\boldsymbol{h}_i) = p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)$. Therefore the KL divergence

$$D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)) \;=\; 0 \tag{52}$$

is zero, thus $\mathcal{F}$ is equal to the log-likelihood $\log p(\boldsymbol{v}_i)$ ($\mathcal{F} = \log p(\boldsymbol{v}_i)$). The **M-step** maximizes $\mathcal{F}$ with respect to the parameters $(\boldsymbol{W}, \boldsymbol{\Psi})$, therefore the M-step maximizes $\int Q(\boldsymbol{h}_i) \log p(\boldsymbol{v}_i \mid \boldsymbol{h}_i) d\boldsymbol{h}_i$.

We next consider again all n samples $\{\boldsymbol{v}\} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$. The *expected reconstruction error* $\mathcal{E}$ for these n data samples is

$$\mathcal{E} \;=\; -\; \frac{1}{n} \sum_{i=1}^{n} \int_{\mathbb{R}^l} Q(\boldsymbol{h}_i) \;\log\left(p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\right)\; d\boldsymbol{h}_i \;=\; \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\log\left(p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\right)\right) \tag{53}$$

and objective to maximize becomes

$$\mathcal{F} \;=\; -\,\mathcal{E} \;-\; \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i)) \;. \tag{54}$$

The M-step requires to minimize $\mathcal{E}$:

$$\mathcal{E} \;=\; \frac{m}{2}\log\left(2\pi\right) \;+\; \frac{1}{2}\log|\boldsymbol{\Psi}| \;+ \tag{55}$$

$$\frac{1}{2\,n}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)^{T}\boldsymbol{\Psi}^{-1}\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)\right)$$

$$= \frac{m}{2}\log\left(2\pi\right) \;+\; \frac{1}{2}\log|\boldsymbol{\Psi}| \;+ \tag{56}$$

$$\frac{1}{2\,n}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\boldsymbol{v}_i^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{v}_i \;-\; 2\,\boldsymbol{v}_i^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\boldsymbol{h}_i \;+\; \boldsymbol{h}_i^{T}\boldsymbol{W}^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\boldsymbol{h}_i\right)$$

$$= \frac{m}{2}\log\left(2\pi\right) \;+\; \frac{1}{2}\log|\boldsymbol{\Psi}| \;+\; \frac{1}{2\,n}\sum_{i=1}^{n}\boldsymbol{v}_i^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{v}_i \tag{57}$$

$$-\;\mathrm{Tr}\left(\boldsymbol{\Psi}^{-1}\boldsymbol{W}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\boldsymbol{h}_i\right)\boldsymbol{v}_i^{T}\right) \;+\; \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{W}^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\boldsymbol{h}_i\boldsymbol{h}_i^{T}\right)\right)$$

$$= \frac{m}{2}\log\left(2\pi\right) \;+\; \frac{1}{2}\log|\boldsymbol{\Psi}| \;+\; \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Psi}^{-1}\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\boldsymbol{v}_i^{T}\right) \tag{58}$$

$$-\;\mathrm{Tr}\left(\boldsymbol{\Psi}^{-1}\boldsymbol{W}\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\mu}_i\boldsymbol{v}_i^{T}\right) \;+\; \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{W}^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{\Sigma} \;+\; \boldsymbol{\mu}_i\boldsymbol{\mu}_i^{T}\right)\right)$$

$$= \frac{1}{2}\left(m\;\log\left(2\pi\right) \;+\; \log|\boldsymbol{\Psi}| \;+\; \mathrm{Tr}\left(\boldsymbol{\Psi}^{-1}\boldsymbol{C}\right)\right. \tag{59}$$

$$\left.-\;2\;\mathrm{Tr}\left(\boldsymbol{\Psi}^{-1}\boldsymbol{W}\boldsymbol{U}^{T}\right) \;+\; \mathrm{Tr}\left(\boldsymbol{W}^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\boldsymbol{S}\right)\right)\;,$$

where Tr gives the trace of a matrix.

The derivatives with respect to the parameters are set to zero for the optimal parameters:

$$\nabla_{\boldsymbol{W}}\mathcal{E} \;=\; -\;\frac{1}{2\,n}\sum_{i=1}^{n}\boldsymbol{\Psi}^{-1}\;\boldsymbol{W}\;\mathrm{E}_{Q}\left(\boldsymbol{h}_i\;\boldsymbol{h}_i^{T}\right) \;+\; \frac{1}{2\,n}\sum_{i=1}^{n}\boldsymbol{\Psi}^{-1}\;\boldsymbol{v}_i\;\mathrm{E}_{Q}^{T}\left(\boldsymbol{h}_i\right) \;=\; \boldsymbol{0} \tag{60}$$

and

$$\nabla_{\boldsymbol{\Psi}}\mathcal{E} \;=\; -\frac{1}{2}\boldsymbol{\Psi}^{-1} + \tag{61}$$

$$\frac{1}{2\,n}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\boldsymbol{\Psi}^{-1}\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)\;\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)^{T}\boldsymbol{\Psi}^{-1}\right) \;=\; \boldsymbol{0}\;.$$

Solving above equations gives:

$$\boldsymbol{W}^{\mathrm{new}} \;=\; \left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\;\mathrm{E}_{\boldsymbol{h}_i|\boldsymbol{v}_i}^{T}\left(\boldsymbol{h}_i\right)\right)\;\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\boldsymbol{h}_i\;\boldsymbol{h}_i^{T}\right)\right)^{-1} \tag{62}$$

and

$$\boldsymbol{\Psi}^{\mathrm{new}} \;=\; \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}^{\mathrm{new}}\boldsymbol{h}_i\right)\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}^{\mathrm{new}}\boldsymbol{h}_i\right)^{T}\right) \;= \tag{63}$$

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\;\boldsymbol{v}_i^{T} \;-\; \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\mathrm{E}_{Q}^{T}\left(\boldsymbol{h}_i\right)\left(\boldsymbol{W}^{\mathrm{new}}\right)^{T} \;-$$

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{W}^{\mathrm{new}}\mathrm{E}_{Q}\left(\boldsymbol{h}_i\right)\boldsymbol{v}_i^{T} \;+\; \boldsymbol{W}^{\mathrm{new}}\frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{Q}\left(\boldsymbol{h}_i\;\boldsymbol{h}_i^{T}\right)\left(\boldsymbol{W}^{\mathrm{new}}\right)^{T}\;.$$

We obtain the following EM updates:

$$\textbf{E-step:} \tag{64}$$

$$\begin{aligned}
\boldsymbol{\mu}_i &= \left(\boldsymbol{I} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{v}_i \,, \\
\boldsymbol{\Sigma} &= \left(\boldsymbol{I} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \,, \\
\mathrm{E}_Q\left(\boldsymbol{h}_i\right) &= \boldsymbol{\mu}_i \\
\mathrm{E}_Q\left(\boldsymbol{h}_i \, \boldsymbol{h}_i^T\right) &= \boldsymbol{\mu}_i \, \boldsymbol{\mu}_i^T + \boldsymbol{\Sigma}
\end{aligned}$$

$$\textbf{M-step:} \tag{65}$$

$$\boldsymbol{W}^{\mathrm{new}} = \left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \, \mathrm{E}^T_{\boldsymbol{h}_i|\boldsymbol{v}_i}\left(\boldsymbol{h}_i\right)\right) \left(\frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\boldsymbol{h}_i \, \boldsymbol{h}_i^T\right)\right)^{-1}$$

$$\begin{aligned}
\boldsymbol{\Psi}^{\mathrm{new}} &= \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \, \boldsymbol{v}_i^T - \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \mathrm{E}^T_Q\left(\boldsymbol{h}_i\right) \left(\boldsymbol{W}^{\mathrm{new}}\right)^T - \\
&\quad \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{W}^{\mathrm{new}} \mathrm{E}_Q\left(\boldsymbol{h}_i\right) \boldsymbol{v}_i^T + \boldsymbol{W}^{\mathrm{new}} \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\boldsymbol{h}_i \, \boldsymbol{h}_i^T\right) \left(\boldsymbol{W}^{\mathrm{new}}\right)^T \,.
\end{aligned} \tag{66}$$

The EM algorithms can be reformulated as:

$$\textbf{E-step:} \tag{67}$$

$$\begin{aligned}
\boldsymbol{\mu}_i &= \left(\boldsymbol{I} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{v}_i \,, \\
\boldsymbol{\Sigma} &= \left(\boldsymbol{I} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \,, \\
\mathrm{E}_Q\left(\boldsymbol{h}_i\right) &= \boldsymbol{\mu}_i \\
\mathrm{E}_Q\left(\boldsymbol{h}_i \, \boldsymbol{h}_i^T\right) &= \boldsymbol{\mu}_i \, \boldsymbol{\mu}_i^T + \boldsymbol{\Sigma}
\end{aligned}$$

$$\textbf{M-step:} \tag{68}$$

$$\boldsymbol{C} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \, \boldsymbol{v}_i^T$$

$$\boldsymbol{U} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \, \mathrm{E}^T_Q\left(\boldsymbol{h}_i\right) \tag{69}$$

$$\boldsymbol{S} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\boldsymbol{h}_i \, \boldsymbol{h}_i^T\right) \tag{70}$$

$$\boldsymbol{W}^{\mathrm{new}} = \boldsymbol{U} \, \boldsymbol{S}^{-1} \tag{71}$$

$$\boldsymbol{\Psi}^{\mathrm{new}} = \boldsymbol{C} - \boldsymbol{U}\boldsymbol{W}^T - \boldsymbol{W}\boldsymbol{U}^T + \boldsymbol{W}\boldsymbol{S}\boldsymbol{W}^T \,. \tag{72}$$

6 The RFN Objective

Our goal is to find a sparse, non-negative representation of the input which extracts structure from the input. A sparse, non-negative representation is desired to code only events or objects that have caused the input. We assume that only few events or objects caused the input, therefore, we aim at sparseness. Furthermore, we do not want to code the degree of absence of events or objects. As the vast majority of events and objects is supposed to be absent, to code for their degree of absence would introduce a high level of random fluctuations.

We aim at extracting structures from the input, therefore generative models are use as they explicitly model input structures. For example factor analysis models the covariance structure of the data. However a generative model cannot enforce sparse, non-negative representation of the input.

The input representation of a generative model is the posterior's mean, median, or mode. Generative models with rectified priors (zero probability for negative values) lead to rectified posteriors. However these posteriors do not have sparse means (they must be positive), that is, they do not yield sparse codes Frey and Hinton [1999]. For example, rectified factor analysis, which rectifies Gaussian priors and selects models using a variational Bayesian learning procedure, does not yield posteriors with sparse means Harva and Kaban [2005, 2007]. A generative model with hidden units $\boldsymbol{h}$ and data $\boldsymbol{v}$ is defined by its prior $p(\boldsymbol{h})$ and its likelihood $p(\boldsymbol{v} \mid \boldsymbol{h})$. The posterior $p(\boldsymbol{h} \mid \boldsymbol{v})$ supplies the input representation of a model by the posterior's mean, median, or mode. However, the posterior depends on the data $\boldsymbol{v}$, therefore sparseness and non-negativity of its means cannot be guaranteed independent of the data. Problem at coding the input by generative models is the data-dependency of the posterior means.

Therefore we use the *posterior regularization method* (*posterior constraint method*) Ganchev et al. [2010], Graca et al. [2009, 2007]. The posterior regularization framework separates model characteristics from data dependent characteristics like the likelihood or posterior constraints. Posterior regularization incorporates data-dependent characteristics as constraints on model posteriors given the observed data, which are difficult to encode via model parameters by Bayesian priors.

A generative model with prior $p(\boldsymbol{h})$ and likelihood $p(\boldsymbol{v} \mid \boldsymbol{h})$ has the full model distribution $p(\boldsymbol{h}, \boldsymbol{v}) = p(\boldsymbol{v} \mid \boldsymbol{h})p(\boldsymbol{h})$. It can be written as $p(\boldsymbol{h}, \boldsymbol{v}) = p(\boldsymbol{h} \mid \boldsymbol{v})p(\boldsymbol{v})$, where $p(\boldsymbol{h} \mid \boldsymbol{v})$ is the model posterior of the hidden variables and $p(\boldsymbol{v})$ is the evidence, that is, the likelihood of the data to be produced by the model. The model family and its parametrization determines which structures are extracted from the data. Typically the model parameters enter the likelihood $p(\boldsymbol{v} \mid \boldsymbol{h})$ and are adjusted to the observed data. For the posterior regularization method, a family $\mathcal{Q}$ of allowed posterior distributions is introduced. $\mathcal{Q}$ is defined by the expectations of constraint features. In our case the posterior means have to be non-negative. Distributions $Q \in \mathcal{Q}$ are called *variational distributions* (see later for using this term). The full variational distribution is $Q(\boldsymbol{h}, \boldsymbol{v}) = Q(\boldsymbol{h} \mid \boldsymbol{v})p_v(\boldsymbol{v})$ with $Q(\boldsymbol{h} \mid \boldsymbol{v}) \in \mathcal{Q}$. The distribution $p_v(\boldsymbol{v})$ is the unknown distribution of observations as determined by the world or the data generation process. This distribution is approximated by samples drawn from the world, namely the training samples. $p(\boldsymbol{h}, \boldsymbol{v})$ *contains all model assumptions like the structures used to model the data, while* $Q(\boldsymbol{h}, \boldsymbol{v})$ *contains all data dependent characteristics including data dependent constraints on the posterior.*

The goal is to achieve $Q(\boldsymbol{h}, \boldsymbol{v}) = p(\boldsymbol{h}, \boldsymbol{v})$, to obtain (1) a desired structure that is extracted from the data and (2) desired code properties. However in general it is to achieve this identity, therefore we want to minimize the distance between these distributions. We use the Kullback-Leibler (KL) divergence Kullback and Leibler [1951] D_{KL} to measure the distance between these distributions. Therefore our objective is $D_{\mathrm{KL}}(Q(\boldsymbol{h}, \boldsymbol{v}) \parallel p(\boldsymbol{h}, \boldsymbol{v}))$. Minimizing this KL divergence (1) extracts the desired structure from the data by increasing the likelihood, that is, $p_v(\boldsymbol{v}) \approx p(\boldsymbol{v})$, and (2) enforces desired code properties by $Q(\boldsymbol{h} \mid \boldsymbol{v}) \approx p(\boldsymbol{h} \mid \boldsymbol{v})$. Thus, the code derived from $Q(\boldsymbol{h} \mid \boldsymbol{v})$ has the desired properties and t extracts the desired input data structures.

We now approximate the KL divergence by approximating the expectation over $p_v(\boldsymbol{v})$ by the empirical mean of samples $\{\boldsymbol{v}\} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ drawn from $p_v(\boldsymbol{v})$:

$$
\begin{aligned}
D_{\mathrm{KL}}(Q(\boldsymbol{h}, \boldsymbol{v}) \parallel p(\boldsymbol{h}, \boldsymbol{v})) \; &= \; \int Q(\boldsymbol{h}, \boldsymbol{v}) \; \log \frac{Q(\boldsymbol{h}, \boldsymbol{v})}{p(\boldsymbol{h}, \boldsymbol{v})} \; d\boldsymbol{h} \; d\boldsymbol{v} \qquad (73) \\
&= \; \int_V p_v(\boldsymbol{v}) \; \int_H Q(\boldsymbol{h} \mid \boldsymbol{v}) \; \log \frac{Q(\boldsymbol{h}, \boldsymbol{v})}{p(\boldsymbol{h}, \boldsymbol{v})} \; d\boldsymbol{h} \; d\boldsymbol{v} \\
&\approx \; \frac{1}{n} \; \sum_{i=1}^{n} \int_H Q(\boldsymbol{h} \mid \boldsymbol{v}_i) \; \log \frac{Q(\boldsymbol{h}, \boldsymbol{v}_i)}{p(\boldsymbol{h}, \boldsymbol{v}_i)} \; d\boldsymbol{h} \\
&= \; \frac{1}{n} \; \sum_{i=1}^{n} \int_H Q(\boldsymbol{h} \mid \boldsymbol{v}_i) \; \log \frac{Q(\boldsymbol{h} \mid \boldsymbol{v}_i)}{p(\boldsymbol{h}, \boldsymbol{v}_i)} \; d\boldsymbol{h} \; + \; \frac{1}{n} \; \sum_{i=1}^{n} \log p_v(\boldsymbol{v}_i) \; .
\end{aligned}
$$

The last term $\frac{1}{n} \sum_{i=1}^{n} \log p_v(\boldsymbol{v}_i)$ neither depends on Q nor on the model, therefore we will neglect it. In the following, we often abbreviate $Q(\boldsymbol{h} \mid \boldsymbol{v}_i)$ by $Q(\boldsymbol{h}_i)$ or write $Q(\boldsymbol{h}_i \mid \boldsymbol{v}_i)$, since the hidden variable is based on the observation $\boldsymbol{v}_i$. Similarly we often write $p(\boldsymbol{h}_i, \boldsymbol{v}_i)$ instead of $p(\boldsymbol{h}, \boldsymbol{v}_i)$ and even more often $p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)$ instead of $p(\boldsymbol{h} \mid \boldsymbol{v}_i)$.

We obtain the objective $\mathcal{F}$ (to be maximized) of the *posterior constraint method* Ganchev et al. [2010], Graca et al. [2009, 2007]:

$$\begin{aligned}
\mathcal{F} &= \frac{1}{n}\sum_{i=1}^{n}\log p(\boldsymbol{v}_i) \;-\; \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)) \qquad (74)\\
&= \frac{1}{n}\sum_{i=1}^{n}\int Q(\boldsymbol{h}_i)\;\log p(\boldsymbol{v}_i)\; d\boldsymbol{h}_i \;-\; \frac{1}{n}\sum_{i=1}^{n}\int Q(\boldsymbol{h}_i)\;\log\frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)}\; d\boldsymbol{h}_i\\
&= -\frac{1}{n}\sum_{i=1}^{n}\int Q(\boldsymbol{h}_i)\;\log\frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i, \boldsymbol{v}_i)}\; d\boldsymbol{h}_i\\
nonumber = &-\frac{1}{n}\sum_{i=1}^{n}\int Q(\boldsymbol{h}_i)\;\log\frac{Q(\boldsymbol{h}_i)}{p(\boldsymbol{h}_i)}\; d\boldsymbol{h}_i \;+\; \frac{1}{n}\sum_{i=1}^{n}\int Q(\boldsymbol{h}_i)\;\log p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\; d\boldsymbol{h}_i \qquad (75)\\
&= \frac{1}{n}\sum_{i=1}^{n}\int Q(\boldsymbol{h}_i)\;\log p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\; d\boldsymbol{h}_i \;-\; \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i))\;.
\end{aligned}$$

The first line is the negative objective of the posterior constraint method while the third line is the negative Eq. (73) without the term $\frac{1}{n}\sum_{i=1}^{n}\log p_v(\boldsymbol{v}_i)$.

$\mathcal{F}$ is the objective in our framework which has to be maximized. Maximizing $\mathcal{F}$ (1) increases the model likelihood $\frac{1}{n}\sum_{i=1}^{n}\log p(\boldsymbol{v}_i)$, (2) finds a proper input representation by small $D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i))$. Thus, the data representation (1) extracts structures from the data as imposed by the generative model while (2) ensuring desired code properties via $Q \in \mathcal{Q}$.

In the variational framework, Q is the variational distribution and $\mathcal{F}$ is called the negative *free energy* Neal and Hinton [1998]. This physical term is used since variational methods were introduced for quantum physics by Richard Feynman Feynman [1972]. The hidden variables can be considered as the fictive causes or explanations of environmental fluctuations Friston [2012].

If $p(\boldsymbol{h} \mid \boldsymbol{v}) \in \mathcal{Q}$, then $Q(\boldsymbol{h} \mid \boldsymbol{v}) = p(\boldsymbol{h} \mid \boldsymbol{v})$ and we obtain the classical EM algorithm. The EM algorithm maximizes the lower bound $\mathcal{F}$ on the log-likelihood as seen at the first line of Eq. (74) and ensures in its E-step $Q(\boldsymbol{h} \mid \boldsymbol{v}) = p(\boldsymbol{h} \mid \boldsymbol{v})$.

7 Generalized Alternating Minimization

Instead of the EM algorithm we use the *Generalized Alternating Minimization (GAM)* algorithm Gunawardana and Byrne [2005] to allow for gradient descent both in the M-step and the E-step. The representation of an input by a generative model is the vector of the mean values of the posterior, that is, the most likely hidden variables that produced the observed data. We have to modify the E-step to enforce variational distributions which lead to sparse codes via zero values of the components of its mean vector. Sparse codes, that is, many components of the mean vector are zero, are obtained by enforcing non-negative means. This rectification is analog to rectified linear units for neural networks, which have enabled sparse codes for neural networks. Therefore the variational distributions are restricted to stem from a family with non-negative constraints on the means. To impose constraints on the posterior is known as the *posterior constraint method* Ganchev et al. [2010], Graca et al. [2009, 2007]. The posterior constraint method maximizes the objective both in the E-step and the M-step. The posterior constraint method is computationally infeasible for our approach, since we assume a large number of hidden units. For models with many hidden units, the maximization in the E-step would take too much time. The posterior constraint method does not support fast implementations on GPUs and stochastic gradients, which we want to allow in order to use mini-batches and dropout regularization.

Therefore we perform only one gradient descent step both in the E-step and in the M-step. Unfortunately, the convergence proofs of the EM algorithm are no longer valid. However we show that our algorithm is a generalized alternating minimization (GAM) method. Gunawardana and Byrne showed that the GAM converges Gunawardana and Byrne [2005] (see also Wu [1983]).

The following GAM convergence Theorem 4 is Proposition 5 in Gunawardana and Byrne [2005] and proves the convergence of the GAM algorithm to a solution that minimizes $-\mathcal{F}$.

Theorem 4 (GAM Convergence Theorem)**.** *Let the point-to-set map* FB *the composition* $\mathrm{B} \circ \mathrm{F}$ *of point-to-set maps* $\mathrm{F} : \mathcal{D} \times \boldsymbol{\Theta} \to \mathcal{D} \times \boldsymbol{\Theta}$ *and* $\mathrm{B} : \mathcal{D} \times \boldsymbol{\Theta} \to \mathcal{D} \times \boldsymbol{\Theta}$*. Suppose that the point-to-set maps* F *and* B *are defined so that*

(1) F *and* B *are closed on* $\mathcal{D}' \times \boldsymbol{\Theta}$

(2) $F(\mathcal{D}' \times \boldsymbol{\Theta}) \subseteq \mathcal{D} \times \boldsymbol{\Theta}$ *and* $B(\mathcal{D}' \times \boldsymbol{\Theta}) \subseteq \mathcal{D} \times \boldsymbol{\Theta}$

Suppose also that F *is such that all* $(Q'_X, \boldsymbol{\theta}') \in \mathrm{F}(Q_X, \boldsymbol{\theta})$ *have* $\boldsymbol{\theta}' = \boldsymbol{\theta}$ *and satisfy*

$$\textit{(GAM.F):} \qquad D_{\mathrm{KL}}(Q'_X \parallel p_{X;\boldsymbol{\theta}}) \;\leq\; D_{\mathrm{KL}}(Q_X \parallel p_{X;\boldsymbol{\theta}})$$

with equality only if

$$\textit{(EQ.F):} \qquad Q_X \;=\; \arg\min_{Q''_X \in \mathcal{D}} D_{\mathrm{KL}}(Q''_X \parallel p_{X;\boldsymbol{\theta}}) \,,$$

with Q_X *being the unique minimizer. Suppose also that the point-to-set map* B *is such that all* $(Q'_X, \boldsymbol{\theta}') \in \mathrm{B}(Q_X, \boldsymbol{\theta})$ *have* $Q'_X = Q_X$ *and satisfy*

$$\textit{(GAM.B):} \qquad D_{\mathrm{KL}}(Q_X \parallel p_{X;\boldsymbol{\theta}'}) \;\leq\; D_{\mathrm{KL}}(Q_X \parallel p_{X;\boldsymbol{\theta}})$$

with equality only if

$$\textit{(EQ.B):} \qquad \boldsymbol{\theta} \;\in\; \arg\min_{\boldsymbol{\xi} \in \boldsymbol{\Theta}} D_{\mathrm{KL}}(Q_X \parallel p_{X;\boldsymbol{\xi}}) \,.$$

Then,

(1) the point-to-set map FB *is closed on* $\mathcal{D}' \times \boldsymbol{\Theta}$

(2) $FB(\mathcal{D}' \times \boldsymbol{\Theta}) \subseteq \mathcal{D} \times \boldsymbol{\Theta}$

and FB *satisfies the GAM and EQ conditions of the GAM convergence theorem, that is, Theorem 3 in Gunawardana and Byrne [2005].*

Proof. See Proposition 5 in Gunawardana and Byrne [2005]. □

The point-to-set mappings allow extended E-step and M-steps without unique iterates. Therefore, Theorem 4 holds for different implementations, different hardware, different precisions of the algorithm under consideration.

For a GAM method to converge, we have to ensure that the objective increases in both the E-step and the M-step. Q is from a constrained family of variational distributions, while the posterior and the full distribution (observation and hidden units) are both derived from a model family. The model family is a parametrized family. For our models (i) the support of the density models does not depend on the parameter and (ii) the density models are continuous in their parameters. GAM convergence requires both (i) and (ii). Furthermore, both the E-step and the M-step must have unique maximizers and they increase the objective if they are not at a maximum point.

The learning rules, that is, the E-step and the M-step are closed maps as they are continuous functions. The objective for the E-step is strict convex in all its parameters for the variational distributions, simultaneously Dredze et al. [2008, 2012]. It is quadratic for the mean vectors on which constraints are imposed. The objective for the M-step is convex in both parameters $\boldsymbol{W}$ and $\boldsymbol{\Psi}^{-1}$ (we sometimes estimate $\boldsymbol{\Psi}$ instead of $\boldsymbol{\Psi}^{-1}$). The objective is quadratic in the loading matrix $\boldsymbol{W}$. For rectifying only, we guarantee unique global maximizers by convex and compact sets for both the family of desired distributions and the set of possible parameters. For this convex optimization problem with one *global* maximum. For rectifying and normalizing, the family of desired distributions is not convex due to equality constraints introduced by the normalization. However we can guarantee *local* unique maximizers.

Summary of the requirements for GAM convergence Theorem 4:

1. the learning rules, that is, the E-step and the M-step, are closed maps,
2. the parameter set is compact,

3. the family of variational distributions is compact (often described by the feasible set of parameters of the variational distributions),
4. the support of the density models does not depend on the parameter,
5. the density models are continuous in the parameters,
6. the E-step has a unique maximizer,
7. the E-step increases the objective if not at the maximizer,
8. the M-step has a unique maximizer (not required by Theorem 4),
9. the M-step increases the objective if not at the maximizer.

The resulting model from the GAM procedure is at a local maximum of the objective given the model family and the family of variational distributions. *The solution minimizes the KL-distance between the family of full variational distributions and full model family*. "Full" means that both the observed and the hidden variables are taken into account, where for the variational distributions the probability of the observations is set to 1. The *desired family* is defined as the set of all probability distributions that assign probability one to the observation. In our case the family of variational distributions is not the desired family since some distributions are excluded by the constraints. Therefore the solution of the GAM optimization does not guarantee stationary points in likelihood Gunawardana and Byrne [2005]. This means that we do not maximize the likelihood but minimize the KL-distance between variational distributions and model.

8 Gradient-based M-step

8.1 Gradient Ascent

The gradients in the M-step are:

$$\nabla_{\boldsymbol{W}} \mathcal{E} \;=\; \frac{1}{2\,n} \sum_{i=1}^{n} \boldsymbol{\Psi}^{-1}\, \boldsymbol{v}_i\, \mathrm{E}_Q^T\left(\boldsymbol{h}_i\right) \;-\; \frac{1}{2\,n} \sum_{i=1}^{n} \boldsymbol{\Psi}^{-1}\, \boldsymbol{W}\, \mathrm{E}_Q\left(\boldsymbol{h}_i\, \boldsymbol{h}_i^T\right)$$

and

$$\nabla_{\boldsymbol{\Psi}} \mathcal{E} \;=\; -\frac{1}{2} \boldsymbol{\Psi}^{-1} \;+\; \frac{1}{2\,n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\boldsymbol{\Psi}^{-1}\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)\, \left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)^T \boldsymbol{\Psi}^{-1}\right) . \tag{76}$$

Alternatively, we can estimate $\boldsymbol{\Psi}^{-1}$ which leads to the derivatives:

$$\nabla_{\boldsymbol{\Psi}^{-1}} \mathcal{E} \;=\; \frac{1}{2} \boldsymbol{\Psi} \;-\; \frac{1}{2\,n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)\, \left(\boldsymbol{v}_i \;-\; \boldsymbol{W}\boldsymbol{h}_i\right)^T\right) . \tag{77}$$

Scaling the gradients leads to:

$$2\, \nabla_{\boldsymbol{W}} \mathcal{E} \;=\; \boldsymbol{\Psi}^{-1}\, \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i\, \mathrm{E}_Q^T\left(\boldsymbol{h}_i\right) \;-\; \boldsymbol{\Psi}^{-1}\, \boldsymbol{W}\, \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\boldsymbol{h}_i\, \boldsymbol{h}_i^T\right) \tag{78}$$

and

$$\begin{aligned} & 2\, \nabla_{\boldsymbol{\Psi}} \mathcal{E} \;= \\ & -\; \boldsymbol{\Psi}^{-1} \;+\; \boldsymbol{\Psi}^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i\, \boldsymbol{v}_i^T \;-\; \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i\, \mathrm{E}_Q^T\left(\boldsymbol{h}_i\right)\, \boldsymbol{W}^T \right. \\ & \left. -\; \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{W}\, \mathrm{E}_Q\left(\boldsymbol{h}_i\right)\, \boldsymbol{v}_i^T \;+\; \boldsymbol{W}\, \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_Q\left(\boldsymbol{h}_i\, \boldsymbol{h}_i^T\right) \boldsymbol{W}^T \right) \boldsymbol{\Psi}^{-1} . \end{aligned} \tag{79}$$

or

$$
\begin{aligned}
& 2\,\nabla_{\boldsymbol{\Psi}^{-1}}\mathcal{E} \;=\; \\
& \boldsymbol{\Psi} \;-\; \left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\,\boldsymbol{v}_i^T \;-\; \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{v}_i\,\mathrm{E}_Q^T\left(\boldsymbol{h}_i\right)\,\boldsymbol{W}^T\right. \\
& \left. -\,\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{W}\,\mathrm{E}_Q\left(\boldsymbol{h}_i\right)\,\boldsymbol{v}_i^T \;+\; \boldsymbol{W}\,\frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_Q\left(\boldsymbol{h}_i\,\boldsymbol{h}_i^T\right)\boldsymbol{W}^T\right)\,.
\end{aligned}
\tag{80}
$$

Only the sums

$$
\boldsymbol{U} \;=\; \frac{1}{n}\,\sum_{i=1}^{n}\boldsymbol{v}_i\mathrm{E}_Q^T\left(\boldsymbol{h}_i\right) \tag{81}
$$

and

$$
\boldsymbol{S} \;=\; \frac{1}{n}\,\sum_{i=1}^{n}\mathrm{E}_Q\left(\boldsymbol{h}_i\,\boldsymbol{h}_i^T\right) \tag{82}
$$

must be computed for both gradients.

$$
\boldsymbol{C} \;=\; \frac{1}{n}\,\sum_{i=1}^{n}\boldsymbol{v}_i\,\boldsymbol{v}_i^T \tag{83}
$$

is the estimated covariance matrix (matrix of second moments for zero mean).

The generalized EM algorithm update rules are:

$$
\boldsymbol{C} \;=\; \frac{1}{n}\,\sum_{i=1}^{n}\boldsymbol{v}_i\,\boldsymbol{v}_i^T \tag{84}
$$

E-step:

$$
\begin{aligned}
\boldsymbol{\mu}_i \;&=\; \boldsymbol{W}^T\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\,\boldsymbol{v}_i \;=\; \left(\boldsymbol{I} \;+\; \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\,\boldsymbol{v}_i\,, \\
\boldsymbol{\Sigma} \;&=\; \boldsymbol{I} \;-\; \boldsymbol{W}^T\,\left(\boldsymbol{W}\,\boldsymbol{W}^T \;+\; \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W} \;=\; \left(\boldsymbol{I} \;+\; \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,, \\
\mathrm{E}_Q\left(\boldsymbol{h}_i\right) \;&=\; \boldsymbol{\mu}_i \\
\mathrm{E}_Q\left(\boldsymbol{h}_i\,\boldsymbol{h}_i^T\right) \;&=\; \boldsymbol{\mu}_i\,\boldsymbol{\mu}_i^T \;+\; \boldsymbol{\Sigma} \\
\boldsymbol{U} \;&=\; \frac{1}{n}\,\sum_{i=1}^{n}\boldsymbol{v}_i\mathrm{E}_Q^T\left(\boldsymbol{h}_i\right) \\
\boldsymbol{S} \;&=\; \frac{1}{n}\,\sum_{i=1}^{n}\mathrm{E}_Q\left(\boldsymbol{h}_i\,\boldsymbol{h}_i^T\right)
\end{aligned}
$$

M-step:

$$
\begin{aligned}
\Delta\boldsymbol{W} \;&=\; \boldsymbol{\Psi}^{-1}\,\boldsymbol{U} \;-\; \boldsymbol{\Psi}^{-1}\,\boldsymbol{W}\,\boldsymbol{S} \\
\Delta\boldsymbol{\Psi} \;&=\; -\,\boldsymbol{\Psi}^{-1} \;+\; \boldsymbol{\Psi}^{-1}\left(\boldsymbol{C} \;-\; \boldsymbol{U}\,\boldsymbol{W}^T \;-\; \boldsymbol{W}\,\boldsymbol{U} \;+\; \boldsymbol{W}\,\boldsymbol{S}\,\boldsymbol{W}^T\right)\boldsymbol{\Psi}^{-1}\,.
\end{aligned}
\tag{85}
$$

8.2 Newton Update

Instead of gradient ascent, we now consider a Newton update step. The Newton update for finding the roots of $\frac{\partial f}{\partial \boldsymbol{v}}$ is

$$
\boldsymbol{v}_{n+1} \;=\; \boldsymbol{v}_n \;-\; \eta\,\boldsymbol{H}^{-1}\,\nabla_{\boldsymbol{v}} f(\boldsymbol{v}_n)\,, \tag{86}
$$

where η is a small step size and $\boldsymbol{H}$ is the Hessian of f with respect to $\boldsymbol{v}$ evaluated at $\boldsymbol{v}_n$. We denote the update direction by

$$
\Delta\boldsymbol{v} \;=\; -\,\boldsymbol{H}^{-1}\,\nabla_{\boldsymbol{v}} f(\boldsymbol{v}_n)\,. \tag{87}
$$

8.2.1 Newton Update of the Loading Matrix

Theorem 5 (Newton Update for Loading Matrix)**.** *The M-step objective $\mathcal{E}$ is quadratic in $\boldsymbol{W}$, thus convex in $\boldsymbol{W}$. The Newton update direction for $\boldsymbol{W}$ in the M-step is*

$$\Delta \boldsymbol{W} \ = \ \boldsymbol{U} \, \boldsymbol{S}^{-1} \ - \ \boldsymbol{W} \ . \tag{88}$$

Proof. The M-step objective is the *expected reconstruction error* $\mathcal{E}$, which is according to Eq. (55)

$$\begin{aligned} \mathcal{E} \ &= \ - \, \frac{1}{n} \, \sum_{i=1}^{n} \int_{\mathbb{R}^l} Q(\boldsymbol{h}_i) \ \log \left(p(\boldsymbol{v}_i \mid \boldsymbol{h}_i) \right) \ d\boldsymbol{h}_i \ = \ \frac{1}{2} \Big(m \ \log \left(2 \pi \right) \ + \ \log |\boldsymbol{\Psi}| \\ &+ \ \mathrm{Tr} \left(\boldsymbol{\Psi}^{-1} \boldsymbol{C} \right) \ - \ 2 \ \mathrm{Tr} \left(\boldsymbol{\Psi}^{-1} \boldsymbol{W} \boldsymbol{U}^T \right) \ + \ \mathrm{Tr} \left(\boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W} \boldsymbol{S} \right) \Big) \ , \end{aligned} \tag{89}$$

where Tr gives the trace of a matrix. This is a quadratic function in $\boldsymbol{W}$, as stated in the theorem.

The Hessian $\boldsymbol{H}_{\boldsymbol{W}}$ of $(2\mathcal{E})$ with respect to $\boldsymbol{W}$ as a vector is:

$$\begin{aligned} \boldsymbol{H}_{\boldsymbol{W}} \ &= \ \frac{\partial \mathrm{vec} \left(2 \, \nabla_{\boldsymbol{W}} \mathcal{E} \right)}{\partial \mathrm{vec}(\boldsymbol{W})^T} \ = \ \frac{\partial \mathrm{vec} \left(- \, \boldsymbol{\Psi}^{-1} \, \boldsymbol{U} \ + \ \boldsymbol{\Psi}^{-1} \, \boldsymbol{W} \, \boldsymbol{S} \right)}{\partial \mathrm{vec}(\boldsymbol{W})^T} \\ &= \ \boldsymbol{S} \ \otimes \ \boldsymbol{\Psi}^{-1} \, , \end{aligned} \tag{90}$$

where $\otimes$ is the Kronecker product of matrices. $\boldsymbol{H}_{\boldsymbol{W}}$ is positive definite, thus the problem is convex in $\boldsymbol{W}$. The inverse of $\boldsymbol{H}_{\boldsymbol{W}}$ is

$$\boldsymbol{H}_{\boldsymbol{W}}^{-1} \ = \ \boldsymbol{S}^{-1} \ \otimes \ \boldsymbol{\Psi} \ . \tag{91}$$

For the product of the inverse Hessian with the gradient we have:

$$\begin{aligned} &\boldsymbol{H}_{\boldsymbol{W}}^{-1} \ \mathrm{vec} \left(- \, \boldsymbol{\Psi}^{-1} \, \boldsymbol{U} \ + \ \boldsymbol{\Psi}^{-1} \, \boldsymbol{W} \, \boldsymbol{S} \right) \ = \ \mathrm{vec} \left(\boldsymbol{\Psi} \ \left(- \, \boldsymbol{\Psi}^{-1} \, \boldsymbol{U} \ + \ \boldsymbol{\Psi}^{-1} \, \boldsymbol{W} \, \boldsymbol{S} \right) \boldsymbol{S}^{-1} \right) \\ &= \mathrm{vec} \left(- \, \boldsymbol{U} \, \boldsymbol{S}^{-1} \ + \ \boldsymbol{W} \right) \ . \end{aligned} \tag{92}$$

If we apply a Newton update, then the update direction for $\boldsymbol{W}$ in the M-step is

$$\Delta \boldsymbol{W} \ = \ \boldsymbol{U} \, \boldsymbol{S}^{-1} \ - \ \boldsymbol{W} \ . \tag{93}$$

□

This is the exact EM update if the step-size η is 1. Since the objective is a quadratic function in $\boldsymbol{W}$, one Newton update would lead to the exact solution.

8.2.2 Newton Update of the Noise Covariance

We define the expected approximation error by

$$\begin{aligned} \boldsymbol{E} \ &= \ \boldsymbol{C} \ - \ \boldsymbol{U} \, \boldsymbol{W}^T \ - \ \boldsymbol{W} \, \boldsymbol{U} \ + \ \boldsymbol{W} \, \boldsymbol{S} \, \boldsymbol{W}^T \\ &= \ \frac{1}{n} \, \sum_{i=1}^{n} \mathrm{E}_Q \left((\boldsymbol{v}_i \ - \ \boldsymbol{W} \boldsymbol{h}_i) \ (\boldsymbol{v}_i \ - \ \boldsymbol{W} \boldsymbol{h}_i)^T \right) \ . \end{aligned} \tag{94}$$

$\boldsymbol{\Psi}$ as parameter.

Theorem 6 (Newton Update for Noise Covariance)**.** *The Newton update direction for $\boldsymbol{\Psi}$ as parameter in the M-step is*

$$\Delta \boldsymbol{\Psi} \ = \ \boldsymbol{E} \ - \ \boldsymbol{\Psi} \ . \tag{95}$$

An update with $\Delta \boldsymbol{\Psi}$ ($\eta = 1$) leads to the minimum of the M-step objective $\mathcal{E}$.

Proof. The M-step objective is the *expected reconstruction error* $\mathcal{E}$, which is according to Eq. (55)

$$\begin{aligned} \mathcal{E} \ &= \ - \, \frac{1}{n} \, \sum_{i=1}^{n} \int_{\mathbb{R}^l} Q(\boldsymbol{h}_i) \ \log \left(p(\boldsymbol{v}_i \mid \boldsymbol{h}_i) \right) \ d\boldsymbol{h}_i \ = \ \frac{1}{2} \Big(m \ \log \left(2 \pi \right) \ + \ \log |\boldsymbol{\Psi}| \\ &+ \ \mathrm{Tr} \left(\boldsymbol{\Psi}^{-1} \boldsymbol{C} \right) \ - \ 2 \ \mathrm{Tr} \left(\boldsymbol{\Psi}^{-1} \boldsymbol{W} \boldsymbol{U}^T \right) \ + \ \mathrm{Tr} \left(\boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W} \boldsymbol{S} \right) \Big) \ , \end{aligned} \tag{96}$$

where Tr gives the trace of a matrix.

Since

$$2 \nabla_{\boldsymbol{\Psi}} \mathcal{E} \;=\; \boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1} \boldsymbol{E} \boldsymbol{\Psi}^{-1} \,, \tag{97}$$

is

$$\boldsymbol{\Psi} \;=\; \boldsymbol{E} \tag{98}$$

the minimum of $\mathcal{E}$ with respect to $\boldsymbol{\Psi}$. Therefore an update with $\Delta \boldsymbol{\Psi} = \boldsymbol{E} - \boldsymbol{\Psi}$ leads to the minimum.

The Hessian $\boldsymbol{H}_{\boldsymbol{\Psi}}$ of $(2\mathcal{E})$ with respect to $\boldsymbol{\Psi}$ as a vector is:

$$\begin{aligned} \boldsymbol{H}_{\boldsymbol{\Psi}} \;&=\; \frac{\partial \mathrm{vec}\left(2 \nabla_{\boldsymbol{\Psi}} \mathcal{E}\right)}{\partial \mathrm{vec}(\boldsymbol{\Psi})^T} \;=\; \frac{\partial \mathrm{vec}\left(\boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1} \boldsymbol{E} \boldsymbol{\Psi}^{-1}\right)}{\partial \mathrm{vec}(\boldsymbol{\Psi})^T} \\ &= -\, \boldsymbol{\Psi}^{-1} \,\otimes\, \boldsymbol{\Psi}^{-1} \;+\; \boldsymbol{\Psi}^{-1} \,\otimes\, \left(\boldsymbol{\Psi}^{-1} \boldsymbol{E} \boldsymbol{\Psi}^{-1}\right) \;+\; \left(\boldsymbol{\Psi}^{-1} \boldsymbol{E} \boldsymbol{\Psi}^{-1}\right) \otimes\, \boldsymbol{\Psi}^{-1} \,. \end{aligned} \tag{99}$$

The expected approximation error $\boldsymbol{E}$ is a sample estimate for $\boldsymbol{\Psi}$, therefore we have $\boldsymbol{\Psi} \approx \boldsymbol{E}$. The Hessian may not be positive definite for some values of $\boldsymbol{E}$, like for small values of $\boldsymbol{E}$. In order to guarantee a positive definite Hessian, more precisely an approximation to it, for minmization, we set

$$\boldsymbol{E} \;=\; \boldsymbol{\Psi} \tag{100}$$

and obtain

$$\boldsymbol{H}_{\boldsymbol{\Psi}} \;=\; \boldsymbol{\Psi}^{-1} \,\otimes\, \boldsymbol{\Psi}^{-1} \,. \tag{101}$$

We derive an approximate Newton update that is very close to the Newton update.

The inverse of the approximated $\boldsymbol{H}_{\boldsymbol{\Psi}}$ is

$$\boldsymbol{H}_{\boldsymbol{\Psi}}^{-1} \;=\; \boldsymbol{\Psi} \,\otimes\, \boldsymbol{\Psi} \,. \tag{102}$$

For the product of the inverse Hessian with the gradient we have:

$$\begin{aligned} &\boldsymbol{H}_{\boldsymbol{\Psi}}^{-1} \,\mathrm{vec}\left(\boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1} \boldsymbol{E} \boldsymbol{\Psi}^{-1}\right) \;=\; \mathrm{vec}\left(\boldsymbol{\Psi} \,\left(\boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1} \boldsymbol{E} \boldsymbol{\Psi}^{-1}\right) \boldsymbol{\Psi}\right) \\ &= \mathrm{vec}\left(\boldsymbol{\Psi} \;-\; \boldsymbol{E}\right) \,. \end{aligned} \tag{103}$$

If we apply a Newton update, then the update direction for $\boldsymbol{\Psi}$ in the M-step is

$$\Delta \boldsymbol{\Psi} \;=\; \boldsymbol{E} \;-\; \boldsymbol{\Psi} \,. \tag{104}$$

This is the exact EM update if the step-size η is 1. □

$\boldsymbol{\Psi}^{-1}$ as parameter.

Theorem 7 (Newton Update for Inverse Noise Covariance)**.** *The M-step objective $\mathcal{E}$ is convex in $\boldsymbol{\Psi}^{-1}$. The Newton update direction for $\boldsymbol{\Psi}^{-1}$ as parameter in the M-step is*

$$\Delta \boldsymbol{\Psi}^{-1} \;=\; \boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1} \,\boldsymbol{E}\, \boldsymbol{\Psi}^{-1} \,. \tag{105}$$

A first order approximation of this Newton direction for $\boldsymbol{\Psi}$ in the M-step is

$$\Delta \boldsymbol{\Psi} \;=\; \boldsymbol{E} \;-\; \boldsymbol{\Psi} \,. \tag{106}$$

An update with $\Delta \boldsymbol{\Psi}$ ($\eta = 1$) leads to the minimum of the M-step objective $\mathcal{E}$.

Proof. The M-step objective is the *expected reconstruction error* $\mathcal{E}$, which is according to Eq. (55)

$$\begin{aligned} \mathcal{E} \;&=\; -\,\frac{1}{n} \sum_{i=1}^{n} \int_{\mathbb{R}^l} Q(\boldsymbol{h}_i) \;\log\left(p(\boldsymbol{v}_i \mid \boldsymbol{h}_i)\right) \,d\boldsymbol{h}_i \;=\; \frac{1}{2}\Big(m \;\log(2\pi) \;+\; \log|\boldsymbol{\Psi}| \\ &+\; \mathrm{Tr}\left(\boldsymbol{\Psi}^{-1} \boldsymbol{C}\right) \;-\; 2\,\mathrm{Tr}\left(\boldsymbol{\Psi}^{-1} \boldsymbol{W} \boldsymbol{U}^T\right) \;+\; \mathrm{Tr}\left(\boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W} \boldsymbol{S}\right)\Big) \,, \end{aligned} \tag{107}$$

where Tr gives the trace of a matrix.

Since

$$2 \nabla_{\boldsymbol{\Psi}^{-1}} \mathcal{E} \;=\; -\,\boldsymbol{\Psi} \;+\; \boldsymbol{E} \tag{108}$$

is

$$\boldsymbol{\Psi} \;=\; \boldsymbol{E} \tag{109}$$

the minimum of $\mathcal{E}$ with respect to $\boldsymbol{\Psi}^{-1}$. Therefore an update with $\Delta\boldsymbol{\Psi} \;=\; \boldsymbol{E} - \boldsymbol{\Psi}$ leads to the minimum.

The Hessian $\boldsymbol{H}_{\boldsymbol{\Psi}^{-1}}$ of $(2\mathcal{E})$ with respect to $\boldsymbol{\Psi}^{-1}$ as a vector is:

$$\boldsymbol{H}_{\boldsymbol{\Psi}^{-1}} \;=\; \frac{\partial \mathrm{vec}\left(2 \nabla_{\boldsymbol{\Psi}^{-1}} \mathcal{E}\right)}{\partial \mathrm{vec}(\boldsymbol{\Psi}^{-1})^T} \;=\; \frac{\partial \mathrm{vec}\left(-\,\boldsymbol{\Psi} \;+\; \boldsymbol{E}\right)}{\partial \mathrm{vec}(\boldsymbol{\Psi}^{-1})^T} \;=\; \boldsymbol{\Psi} \;\otimes\; \boldsymbol{\Psi}\,. \tag{110}$$

Since the Hessian is positive definite, the E-step objective $\mathcal{E}$ is convex in $\boldsymbol{\Psi}^{-1}$, which is the first statement of the theorem.

The inverse of $\boldsymbol{H}_{\boldsymbol{\Psi}^{-1}}$ is

$$\boldsymbol{H}^{-1}_{\boldsymbol{\Psi}^{-1}} \;=\; \boldsymbol{\Psi}^{-1} \;\otimes\; \boldsymbol{\Psi}^{-1}\,. \tag{111}$$

For the product of the inverse Hessian with the gradient we have:

$$\begin{aligned} &\boldsymbol{H}^{-1}_{\boldsymbol{\Psi}^{-1}} \,\mathrm{vec}\left(-\,\boldsymbol{\Psi} \;+\; \boldsymbol{E}\right) \;=\; \mathrm{vec}\left(\boldsymbol{\Psi}^{-1} \;\left(-\,\boldsymbol{\Psi} \;+\; \boldsymbol{E}\right) \boldsymbol{\Psi}^{-1}\right) \\ &= \mathrm{vec}\left(-\,\boldsymbol{\Psi}^{-1} \;+\; \boldsymbol{\Psi}^{-1}\,\boldsymbol{E}\,\boldsymbol{\Psi}^{-1}\right)\,. \end{aligned} \tag{112}$$

If we apply a Newton update, then the update direction for $\boldsymbol{\Psi}^{-1}$ in the M-step is

$$\Delta\boldsymbol{\Psi}^{-1} \;=\; \boldsymbol{\Psi}^{-1} \;-\; \boldsymbol{\Psi}^{-1}\,\boldsymbol{E}\,\boldsymbol{\Psi}^{-1}\,. \tag{113}$$

We now can approximate the update for $\boldsymbol{\Psi}$ by the first terms of the Taylor expansion:

$$\boldsymbol{\Psi} \;+\; \Delta\boldsymbol{\Psi} \;=\; \left(\boldsymbol{\Psi}^{-1} \;+\; \Delta\boldsymbol{\Psi}^{-1}\right)^{-1} \;\approx\; \boldsymbol{\Psi} \;-\; \boldsymbol{\Psi}\,\Delta\boldsymbol{\Psi}^{-1}\,\boldsymbol{\Psi}\,. \tag{114}$$

We obtain for the update of $\boldsymbol{\Psi}$

$$\Delta\boldsymbol{\Psi} \;=\; -\,\boldsymbol{\Psi}\,\Delta\boldsymbol{\Psi}^{-1}\,\boldsymbol{\Psi} \;=\; \boldsymbol{E} \;-\; \boldsymbol{\Psi}\,. \tag{115}$$

This is the exact EM update if the step-size η is 1. □

The Newton update derived from $\boldsymbol{\Psi}^{-1}$ as parameter is the Newton update for $\boldsymbol{\Psi}$. Consequently, the Newton direction for both $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}^{-1}$ is in the M-step

$$\Delta\boldsymbol{\Psi} \;=\; \boldsymbol{E} \;-\; \boldsymbol{\Psi}\,. \tag{116}$$

9 Gradient-based E-Step

9.1 Motivation for Rectifying and Normalization Constraints

The representation of data vector $\boldsymbol{v}$ by the model is the variational mean vector $\boldsymbol{\mu}_q$. In order to obtain sparse codes we want to have non-negative $\boldsymbol{\mu}_q$. We enforce non-negative mean values by constraints and optimize by projected Newton methods and by gradient projection methods. Non-negative constraints correspond to rectifying in the neural network field. Therefore we aim to construct sparse codes in analogy to the rectified linear units used for neural networks.

We constrain the variational distributions to the family of normal distributions with non-negative mean components. Consequently we introduce non-negative or **rectifying constraints**:

$$\boldsymbol{\mu} \;\geq\; \mathbf{0}\,, \tag{117}$$

where the inequality "$\geq$" holds component-wise.

However generative models with many coding units face a problem. They tend to *explain away small and rare signals by noise*. For many coding units, model selection algorithms prefer models with coding units which do not have variation and, therefore, are removed from the model. Other coding units hardly contribute to explain the observations. The likelihood is larger if small and rare signals are explained by noise, than the likelihood if coding units are use to explain such signals. Coding units without variance are kept on their default values, where they have maximal contribution to the likelihood. If they are used for coding, they deviate from their maximal values for each sample. In accumulation these deviations decrease the likelihood more than it is increased by explaining small or rare signals. For our RFN models the problem can become severe, since we aim at models with up to several tens of thousands of coding units. To avoid the explaining away problem, we enforce the selected models to use all their coding units on an equal level. We do that by keeping the variation of each noise-free coding unit across the training set at one. Consequently, we introduce a **normalization constraint** for each coding unit $1 \leq j \leq l$:

$$\frac{1}{n} \sum_{i=1}^{n} \mu_{ij}^2 \ = \ 1 \ . \tag{118}$$

This constraint means that the noise-free part of each coding unit has variance one across samples.

We will derive methods to increase the objective in the E-step both for only rectifying constraints and for rectifying and normalization constraints. These methods ensure to reduce the objective in the E-step to guarantee convergence via the GAM theory. The resulting model from the GAM procedure is at a local maximum of the objective given the model family and the family of variational distributions. *The solution minimizes the KL-distance between the family of full variational distributions and full model family*. "Full" means that both the observed and the hidden variables are taken into account.

9.2 The Full E-step Objective

The E-step maximizes $\mathcal{F}$ with respect to the variational distribution Q, therefore the E-step minimizes the Kullback-Leibler divergence (KL-divergence) Kullback and Leibler [1951] $D_{\mathrm{KL}}(Q(\boldsymbol{h}) \parallel p(\boldsymbol{h} \mid \boldsymbol{v}))$. The KL-divergence between Q and p is

$$D_{\mathrm{KL}}(Q \parallel p) \ = \ \int Q(\boldsymbol{h}) \ \log \frac{Q(\boldsymbol{h})}{p(\boldsymbol{h} \mid \boldsymbol{v})} \ d\boldsymbol{h} \ . \tag{119}$$

Rectifying constraints introduce non-negative constraints. The minimization with respect to $Q(\boldsymbol{h}_i)$ gives the constraint minimization problem:

$$\begin{aligned} \min_{Q(\boldsymbol{h}_i)} \ & \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)) \\ \text{s.t.} \ & \forall_i : \ \boldsymbol{\mu}_i \ \geq \ \boldsymbol{0} \ , \end{aligned} \tag{120}$$

where $\boldsymbol{\mu}_i$ is the mean vector of $Q(\boldsymbol{h}_i)$.

Rectifying and normalizing constraints introduce non-negative constraints and equality constraints. The minimization with respect to $Q(\boldsymbol{h}_i)$ gives the constraint minimization problem:

$$\begin{aligned} \min_{Q(\boldsymbol{h}_i)} \ & \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(Q(\boldsymbol{h}_i) \parallel p(\boldsymbol{h}_i \mid \boldsymbol{v}_i)) \\ \text{s.t.} \ & \forall_i : \ \boldsymbol{\mu}_i \ \geq \ \boldsymbol{0} \ , \\ & \forall_j : \ \frac{1}{n} \ \sum_{i=1}^{n} \mu_{ij}^2 \ = \ 1 \ , \end{aligned} \tag{121}$$

where $\boldsymbol{\mu}_i$ is the mean vector of $Q(\boldsymbol{h}_i)$.

First we consider the families from which the model and from which the variational distributions stem. The posterior of the model with Gaussian prior $p(\boldsymbol{h})$ is Gaussian (see Section 5):

$$p(\boldsymbol{h} \mid \boldsymbol{v}) \ \sim \ (2\pi)^{-\frac{l}{2}} \ |\boldsymbol{\Sigma}_p|^{-\frac{1}{2}} \ \exp\left(-\frac{1}{2} \ (\boldsymbol{h} \ - \ \boldsymbol{\mu}_p)^T \ \boldsymbol{\Sigma}_p^{-1} \ (\boldsymbol{h} \ - \ \boldsymbol{\mu}_p) \right) \ . \tag{122}$$

To be as close as possible to the posterior distribution, we restrict Q to be from a Gaussian family:

$$Q(\boldsymbol{h}) \;\sim\; (2\pi)^{-\frac{l}{2}} \;|\boldsymbol{\Sigma}_q|^{-\frac{1}{2}} \;\exp\left(-\frac{1}{2}\,(\boldsymbol{h} \;-\; \boldsymbol{\mu}_q)^T\;\boldsymbol{\Sigma}_q^{-1}\;(\boldsymbol{h} \;-\; \boldsymbol{\mu}_q)\right)\,. \tag{123}$$

For Gaussians, the Kullback-Leibler divergence between Q and p is

$$\begin{aligned} &D_{\mathrm{KL}}(Q \parallel p) \;= \\ &\frac{1}{2}\left\{\mathrm{Tr}\left(\boldsymbol{\Sigma}_p^{-1}\;\boldsymbol{\Sigma}_q\right) \;+\; (\boldsymbol{\mu}_p \;-\; \boldsymbol{\mu}_q)^T\;\boldsymbol{\Sigma}_p^{-1}\;(\boldsymbol{\mu}_p \;-\; \boldsymbol{\mu}_q) \;-\; l \;-\; \ln\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_p|}\right\}\,. \end{aligned} \tag{124}$$

This Kullback-Leibler divergence is convex in the mean vector $\boldsymbol{\mu}_q$ and the covariance matrix $\boldsymbol{\Sigma}_q$ of Q, simultaneously Dredze et al. [2008, 2012].

We now minimize Eq. (124) with respect to Q. For the moment we do not care about the constraints introduced by non-negativity and by normalization. Eq. (124) has a quadratic form in $\boldsymbol{\mu}_q$, where $\boldsymbol{\Sigma}_q$ does not enter, and terms in $\boldsymbol{\Sigma}_q$, where $\boldsymbol{\mu}_q$ does not enter. Therefore we can separately minimize for $\boldsymbol{\Sigma}_q$ and for $\boldsymbol{\mu}_q$.

For the minimization with respect to $\boldsymbol{\Sigma}_q$, we require

$$\frac{\partial}{\partial\boldsymbol{\Sigma}_q}\mathrm{Tr}\left(\boldsymbol{\Sigma}_p^{-1}\;\boldsymbol{\Sigma}_q\right) \;=\; \boldsymbol{\Sigma}_p^{-T} \tag{125}$$

and

$$\frac{\partial}{\partial\boldsymbol{\Sigma}_q}\ln|\boldsymbol{\Sigma}_q| \;=\; \boldsymbol{\Sigma}_q^{-T}\,. \tag{126}$$

For optimality the derivative of the objective $D_{\mathrm{KL}}(Q \parallel p)$ with respect to $\boldsymbol{\Sigma}_q$ must be zero:

$$\frac{\partial}{\partial\boldsymbol{\Sigma}_q}D_{\mathrm{KL}}(Q \parallel p) \;=\; \frac{1}{2}\,\boldsymbol{\Sigma}_p^{-T} \;-\; \frac{1}{2}\,\boldsymbol{\Sigma}_q^{-T} \;=\; \mathbf{0}\,. \tag{127}$$

This gives

$$\boldsymbol{\Sigma} \;=\; \boldsymbol{\Sigma}_q \;=\; \boldsymbol{\Sigma}_p\,. \tag{128}$$

We often drop the index q since for $1 \leq i \leq n$ all covariance matrices $\boldsymbol{\Sigma}_q$ are equal to $\boldsymbol{\Sigma}_p$.

The mean vector $\boldsymbol{\mu}_q$ of Q is the solution of the minimization problem:

$$\min_{\boldsymbol{\mu}} \;\; \frac{1}{2}\;(\boldsymbol{\mu}_p \;-\; \boldsymbol{\mu})^T\;\boldsymbol{\Sigma}_p^{-1}\;(\boldsymbol{\mu}_p \;-\; \boldsymbol{\mu}) \tag{129}$$

which is equivalent to

$$\min_{\boldsymbol{\mu}} \;\; \frac{1}{2}\,\boldsymbol{\mu}^T\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\mu} \;-\; \boldsymbol{\mu}_p^T\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\mu}\,. \tag{130}$$

The derivative and the Hessian of this objective is:

$$\frac{\partial}{\partial\boldsymbol{\mu}}D_{\mathrm{KL}}(Q \parallel p) \;=\; \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu} \;-\; \boldsymbol{\mu}_p)\,, \tag{131}$$

$$\frac{\partial^2}{\partial^2\boldsymbol{\mu}}D_{\mathrm{KL}}(Q \parallel p) \;=\; \boldsymbol{\Sigma}_p^{-1}\,. \tag{132}$$

9.3 E-step for Mean with Rectifying Constraints

9.3.1 The E-Step Minimization Problem

Rectifying is realized by non-negative constraints. The mean vector $\boldsymbol{\mu}_q$ of Q is the solution of the minimization problem:

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \;\; \frac{1}{2}\,(\boldsymbol{\mu} \;-\; \boldsymbol{\mu}_p)^T\;\boldsymbol{\Sigma}_p^{-1}\,(\boldsymbol{\mu} \;-\; \boldsymbol{\mu}_p) \\ &\text{s.t.} \;\; \boldsymbol{\mu} \;\geq\; \mathbf{0}\,. \end{aligned} \tag{133}$$

This is a convex quadratic minimization problem with non-negativity constraints (convex feasible set).

If $\boldsymbol{\lambda}$ is the Lagrange multiplier for the constraints, then the dual is

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \frac{1}{2}\, \boldsymbol{\lambda}^T \boldsymbol{\Sigma}_p \boldsymbol{\lambda} \,+\, \boldsymbol{\mu}_p^T \boldsymbol{\lambda} \\ \text{s.t.} \quad & \boldsymbol{\lambda} \,\geq\, \mathbf{0}\,. \end{aligned} \tag{134}$$

The Karush-Kuhn-Tucker conditions require for the optimal solution for each component $1 \leq j \leq l$:

$$\lambda_j \, \mu_j \;=\; 0\,. \tag{135}$$

Further the derivative of the Lagrangian with respect to $\boldsymbol{\mu}$ gives

$$\boldsymbol{\Sigma}_p^{-1} \boldsymbol{\mu} \,-\, \boldsymbol{\Sigma}_p^{-1} \boldsymbol{\mu}_p \,-\, \boldsymbol{\lambda} \,=\, \mathbf{0} \tag{136}$$

which can be written as

$$\boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p \,-\, \boldsymbol{\Sigma}_p \, \boldsymbol{\lambda} \,=\, \mathbf{0}\,. \tag{137}$$

This minimization problem cannot be solved directly. Therefore we perform a gradient projection or projected Newton step to decrease the objective.

9.3.2 The Projection onto the Feasible Set

To decrease the objective, we perform a gradient projection or a projected Newton step. We will base our algorithms on *Euclidean least distance projections*. If projected onto convex sets, these projections do not increase distances. The Euclidean projection onto the feasible set is denoted by P, that is, the map that takes $\boldsymbol{\mu}_p$ to its nearest point $\boldsymbol{\mu}$ (in the L^2-norm) in the feasible set.

For rectifying constraints, the projection P (Euclidean least distance projection) of $\boldsymbol{\mu}_p$ onto the convex feasible set is given by the solution of the convex optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \quad & \frac{1}{2}\, (\boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p)^T \, (\boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p) \\ \text{s.t.} \quad & \boldsymbol{\mu} \,\geq\, \mathbf{0}\,. \end{aligned} \tag{138}$$

The following Theorem 8 shows that update Eq. (139) is the projection P defined by optimization problem Eq. (138).

Theorem 8 (Projection: Rectifying)**.** *The solution to optimization problem Eq.* (138)*, which defines the Euclidean least distance projection, is*

$$\mu_j \;=\; [\mathrm{P}(\boldsymbol{\mu}_p)]_j \;=\; \begin{cases} 0 & \text{for} \quad (\mu_p)_j \,\leq\, 0 \\ (\mu_p)_j & \text{for} \quad (\mu_p)_j \,>\, 0 \end{cases} \tag{139}$$

Proof. For the projection we have the minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \quad & \frac{1}{2}\, (\boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p)^T \, (\boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p) \\ \text{s.t.} \quad & \boldsymbol{\mu} \,\geq\, \mathbf{0}\,. \end{aligned} \tag{140}$$

The Lagrangian L with multiplier $\boldsymbol{\lambda} \geq \mathbf{0}$ is

$$L \;=\; \frac{1}{2}\, (\boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p)^T \, (\boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p) \,-\, \boldsymbol{\lambda}^T \, \boldsymbol{\mu}\,. \tag{141}$$

The derivative with respect to $\boldsymbol{\mu}$ is

$$\frac{\partial L}{\partial \boldsymbol{\mu}} \;=\; \boldsymbol{\mu} \,-\, \boldsymbol{\mu}_p \,-\, \boldsymbol{\lambda} \,=\, \mathbf{0}\,. \tag{142}$$

The Karush-Kuhn-Tucker (KKT) conditions require for the optimal solution that for each constraint j:

$$\lambda_j \, \mu_j \;=\; 0\,. \tag{143}$$

If $0 < (\mu_p)_j$ then Eq. (142) requires $0 < \mu_j$ because the Lagrangian λ_j is larger than or equal to zero: $0 \leq \lambda_j$. From the KKT conditions Eq. (143) follows that $\lambda_j = 0$ and, therefore, $0 < \mu_j = (\mu_p)_j$. If $(\mu_p)_j < 0$ then $0 < \mu_j - (\mu_p)_j$, because the constraints of the primal problem require $0 \leq \mu_j$. From Eq. (142) follows that $0 < \lambda_j$. From the KKT conditions Eq. (143) follows that $(\mu_p)_j = 0$ and $0 < \lambda_j = -(\mu_p)_j$. If $(\mu_p)_j = 0$, then Eq. (142) and the KKT conditions Eq. (143) lead to $(\mu_p)_j = \mu_j = \lambda_j = 0$.

Therefore the solution of problem Eq. (138) is

$$\mu_j \ = \ \begin{cases} (\mu_p)_j & \text{for} \quad (\mu_p)_j > 0 \ \text{ and } \ \lambda_j = 0 \\ 0 & \text{for} \quad (\mu_p)_j \leq 0 \ \text{ and } \ \lambda_j = -(\mu_p)_j \end{cases} \ . \tag{144}$$

This finishes the proof. □

9.4 E-step for Mean with Rectifying and Normalizing Constraints

9.4.1 The E-Step Minimization Problem

If we also consider normalizing constraints, then we have to minimize all KL-divergences simultaneously. The normalizing constraints connect the single optimization problems for each sample $\boldsymbol{v}_i$. For the E-step, we obtain the minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mu}_i} \ & \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\mu}_i \ - \ (\boldsymbol{\mu}_p)_i)^T \ \boldsymbol{\Sigma}_p^{-1} \ (\boldsymbol{\mu}_i \ - \ (\boldsymbol{\mu}_p)_i) \\ \text{s.t.} \ & \forall_i : \ \boldsymbol{\mu}_i \ \geq \ \mathbf{0} \quad , \quad \forall_j : \ \frac{1}{n} \sum_{i=1}^{n} \mu_{ij}^2 \ = \ 1 \ . \end{aligned} \tag{145}$$

The "$\geq$"-sign is meant component-wise. The l equality constraints lead to non-convex feasible sets. The solution to this optimization problem are the means vectors $\boldsymbol{\mu}_i$ of $Q(\boldsymbol{h}_i)$.

Generalized Reduced Gradient. The equality constraints can be solved for one variable which is then inserted into the objective. The equality constraint gives for each $1 \leq j \leq l$:

$$\mu_{1j}^2 \ = \ n \ - \ \sum_{i=2}^{n} \mu_{ij}^2 \quad \text{or} \quad \mu_{1j} \ = \ \sqrt{n \ - \ \sum_{i=2}^{n} \mu_{ij}^2} \ . \tag{146}$$

These equations can be inserted into the objective and, thereby, we remove the variables μ_{1j}. We have to ensure that the μ_{1j} exist by

$$\sum_{i=2}^{n} \mu_{ij}^2 \ \leq \ n \ . \tag{147}$$

These constraints define a convex set feasible set. To solve the each equality constraints for a variable and insert it into the objective is called *generalized reduced gradient* method Abadie and Carpentier [1969]. For solving the reduced problem, we can use methods for constraint optimization were we now ensure a convex feasible set. These methods solve the original problem Eq. (145). We only require an improvement of the objective with a feasible value. For the reduced problem, we perform one step of a *gradient projection method*.

Gradient Projection Methods. Also for the original problem Eq. (145), *gradient projection methods* can be used. The gradient projection method has been generalized by Rosen to *non-linear constraints* Rosen [1961] and was later improved by Haug and Arora [1979]. The gradient projection algorithm of Rosen works for *non-convex feasible sets*. The idea is to linearize the nonlinear constraints and solve the problem. Subsequently a restoration move brings the solution back to the constraint boundaries.

9.4.2 The Projection onto the Feasible Set

To decrease the objective, we perform a gradient projection, a projected Newton step, or a step of the generalized reduced method. We will base our algorithms on *Euclidean least distance projections*. If projected onto convex sets, these projections do not increase distances. The Euclidean projection onto the feasible set is denoted by P, that is, the map that simultaneously takes $\{(\boldsymbol{\mu}_p)_i\}$ to the nearest points $\{\boldsymbol{\mu}_i\}$ (in the L^2-norm) in the feasible set.

For rectifying and normalizing constraints the projection (Euclidean least distance projection) of $\{(\boldsymbol{\mu}_p)_i\}$ onto the **non-convex** feasible set leads to the optimization problem

$$\begin{aligned} \min_{\boldsymbol{\mu}_i} \;\; & \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\mu}_i \;-\; (\boldsymbol{\mu}_p)_i)^T \, (\boldsymbol{\mu}_i \;-\; (\boldsymbol{\mu}_p)_i) \\ \text{s.t.} \;\; & \forall_i : \; \boldsymbol{\mu}_i \;\geq\; \mathbf{0} \,, \\ & \forall_j : \; \frac{1}{n} \sum_{i=1}^{n} \mu_{ij}^2 \;=\; 1 \,. \end{aligned} \tag{148}$$

By using $(\boldsymbol{\mu}_i - (\boldsymbol{\mu}_p)_i)^T \, (\boldsymbol{\mu}_i - (\boldsymbol{\mu}_p)_i) = \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i - 2\boldsymbol{\mu}_i^T (\boldsymbol{\mu}_p)_i + (\boldsymbol{\mu}_p)_i^T (\boldsymbol{\mu}_p)_i$, we see that the objective contains the sum $\sum_{ij} \mu_{ij}^2$. The constraints enforce this sum to be constant. Therefore inserting the equality constraints into the objective, optimization problem Eq. (148) is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\mu}_i} \;\; & -\; \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\mu}_i^T \, (\boldsymbol{\mu}_p)_i \\ \text{s.t.} \;\; & \forall_i : \; \boldsymbol{\mu}_i \;\geq\; \mathbf{0} \,, \\ & \forall_j : \; \frac{1}{n} \sum_{i=1}^{n} \mu_{ij}^2 \;=\; 1 \,. \end{aligned} \tag{149}$$

The following Theorem 9 shows that updates Eq. (150) and Eq. (151) form the projection defined by optimization problem Eq. (148).

Theorem 9 (Projection: Rectifying and Normalizing)**.** *If at least one $(\mu_p)_{ij}$ is positive for $1 \leq j \leq l$, then the solution to optimization problem Eq.* (148)*, which defines the Euclidean least distance projection, is*

$$\begin{aligned} \hat{\mu}_{ij} \;&=\; \begin{cases} 0 & \text{for} \quad (\mu_p)_{ij} \;\leq\; 0 \\ (\mu_p)_{ij} & \text{for} \quad (\mu_p)_{ij} \;>\; 0 \end{cases} \\ \mu_{ij} \;&=\; \left[\mathrm{P}((\boldsymbol{\mu}_p)_i)\right]_j \;=\; \frac{\hat{\mu}_{ij}}{\sqrt{\frac{1}{n} \, \sum_{i=1}^{n} \hat{\mu}_{ij}^2}} \,. \end{aligned} \tag{150}$$

If all $(\mu_p)_{ij}$ are non-positive for $1 \leq j \leq l$, then the optimization problem Eq. (148) *has the solution*

$$\mu_{ij} \;=\; \begin{cases} \sqrt{n} & \text{for} \quad j \;=\; \arg\max_{\hat{j}} \{(\mu_p)_{i\hat{j}}\} \\ 0 & \text{otherwise} \end{cases} \,. \tag{151}$$

Proof. In the following we show that updates Eq. (150) and Eq. (150) are the projection onto the feasible set. For the projection of $\{(\boldsymbol{\mu}_p)_i\}$ onto the feasible set, we have the minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mu}_i} \;\; & \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\mu}_i \;-\; (\boldsymbol{\mu}_p)_i)^T \, (\boldsymbol{\mu}_i \;-\; (\boldsymbol{\mu}_p)_i) \\ \text{s.t.} \;\; & \forall_i : \; \boldsymbol{\mu}_i \;\geq\; \mathbf{0} \,, \\ & \forall_j : \; \frac{1}{n} \sum_{i=1}^{n} \mu_{ij}^2 \;=\; 1 \,. \end{aligned} \tag{152}$$

The feasible set is non-convex because of the quadratic equality constraint. The Lagrangian with multiplier $\boldsymbol{\lambda} \geq \mathbf{0}$ is

$$L \;=\; \frac{1}{n}\,\sum_{i=1}^{n}(\boldsymbol{\mu}_i \;-\; (\boldsymbol{\mu}_p)_i)^T\,(\boldsymbol{\mu}_i \;-\; (\boldsymbol{\mu}_p)_i) \;-\; \sum_{i=1}^{n}\boldsymbol{\lambda}_i^T\,\boldsymbol{\mu}_i \tag{153}$$
$$+\;\sum_j \tau_j\,\left(\frac{1}{n}\,\sum_{i=1}^{n}\mu_{ij}^2 \;-\; 1\right)\,.$$

The Karush-Kuhn-Tucker (KKT) conditions require for the optimal solution:

$$\lambda_{ij}\,\mu_{ij} \;=\; 0 \quad \text{and} \quad \tau_j\,\left(\frac{1}{n}\,\sum_{i=1}^{n}\mu_{ij}^2 \;-\; 1\right) \;=\; 0\,. \tag{154}$$

The derivative of L with respect to μ_{ij} is

$$\frac{\partial L}{\partial \mu_{ij}} \;=\; \frac{2}{n}\,(\mu_{ij} \;-\; (\mu_p)_{ij}) \;-\; \lambda_{ij} \;+\; \frac{2}{n}\,\tau_j\,\mu_{ij} \;=\; 0\,. \tag{155}$$

We multiply this equation by μ_{ij} and obtain:

$$\frac{2}{n}\,(\mu_{ij}^2 \;-\; (\mu_p)_{ij}\,\mu_{ij}) \;-\; \lambda_{ij}\,\mu_{ij} \;+\; \frac{2}{n}\,\tau_j\,\mu_{ij}^2 \;=\; 0\,. \tag{156}$$

The KKT conditions give $\lambda_{ij}\mu_{ij} = 0$, therefore this term can be removed from the equation. Next we sum over i:

$$\frac{2}{n}\,\sum_{i=1}^{n}\left(\mu_{ij}^2 \;-\; (\mu_p)_{ij}\,\mu_{ij}\right) \;+\; \frac{2}{n}\,\sum_{i=1}^{n}\tau_j\,\mu_{ij}^2 \;=\; 0\,. \tag{157}$$

Using the equality constraint $1/n\,\sum_{i=1}^{n}\mu_{ij}^2 = 1$ and dividing by 2 and gives:

$$1 \;-\; \frac{1}{n}\,\sum_{i=1}^{n}(\mu_p)_{ij}\,\mu_{ij} \;+\; \tau_j \;=\; 0\,. \tag{158}$$

Solving for τ_j leads to:

$$\tau_j \;=\; \frac{1}{n}\,\sum_{i=1}^{n}(\mu_p)_{ij}\,\mu_{ij} \;-\; 1\,. \tag{159}$$

We insert τ_j into Eq. (155)

$$-\,(\mu_p)_{ij} \;-\; \frac{n}{2}\lambda_{ij} \;+\; \left(\frac{1}{n}\,\sum_{s=1}^{n}(\mu_p)_{sj}\,\mu_{sj}\right)\,\mu_{ij} \;=\; 0\,. \tag{160}$$

We immediately see, that if $\mu_{ij} = 0$ then $(\mu_p)_{ij} = -\frac{n}{2}\lambda_{ij} < 0$. Therefore we can assume $\mu_{ij} > 0$. Multiplying Eq. (160) with μ_{ij} and using the KKT conditions gives

$$-\,(\mu_p)_{ij}\,\mu_{ij} \;+\; \left(\frac{1}{n}\,\sum_{s=1}^{n}(\mu_p)_{sj}\,\mu_{sj}\right)\,\mu_{ij}^2 \;=\; 0\,. \tag{161}$$

Therefore $(\mu_p)_{ij}\mu_{ij}$ and $\frac{1}{n}\sum_{s=1}^{n}(\mu_p)_{sj}\mu_{sj}$ have the same sign or $\mu_{ij} = 0$. Since $0 \leq \mu_{ij}$, we deduce that $(\mu_p)_{ij}$ and $\frac{1}{n}\sum_{s=1}^{n}(\mu_p)_{sj}\mu_{sj}$ have the same sign or $\mu_{ij} = 0$. Since the sum is independent of i, all $(\mu_p)_{ij}$ with $\mu_{ij} > 0$ have the same sign for $1 \leq i \leq n$. Solving Eq. (160) for μ_{ij} gives

$$\mu_{ij} \;=\; \frac{(\mu_p)_{ij} \;+\; \frac{n}{2}\lambda_{ij}}{\frac{1}{n}\,\sum_{s=1}^{n}(\mu_p)_{sj}\,\mu_{sj}}\,. \tag{162}$$

I. If all $(\mu_p)_{ij}$ are non-positive for $1 \leq j \leq l$, then the sum $\frac{1}{n}\sum_{s=1}^{n}(\mu_p)_{sj}\mu_{sj}$ is negative. From the first order derivative of the Lagrangian in Eq. (155), we can compute the second order derivative

$$\frac{\partial^2 L}{\partial \mu_{ij}\partial \mu_{ij}} \;=\; \frac{2}{n} \;+\; \frac{2}{n}\,\tau_j \;=\; 2\,\sum_{i=1}^{n}(\mu_p)_{ij}\,\mu_{ij} \;<\; 0\,. \tag{163}$$

We inserted the expression of Eq. (159) for τ_j. Since all mixed second order derivatives are zero, the (projected) Hessian of the Lagrangian is diagonal with negative entries. Therefore it is strict negative definite. Thus, the second order necessary conditions cannot be fulfilled. The minimum is a border point of the constraints.

For each j for which all $(\mu_p)_{ij}$ are non-positive for $1 \leq j \leq l$, optimization problem Eq. (149) defines a plane that has a normal vector in the positive orthant (hyperoctant). For such a j the corresponding equality constraint defines a hypersphere. Minimization means that the plane containing the solution is parallel to the original plane and should be as close to the origin as possible. If we move the plane parallel from the origin into the positive orthant, then the first intersection with the hypersphere is

$$\mu_{ij} \;=\; \begin{cases} \sqrt{n} & \text{for} \quad j \;=\; \arg\max_{\hat{j}}\{(\mu_p)_{i\hat{j}}\} \\ 0 & \text{otherwise} \end{cases}\,. \tag{164}$$

This is the solution for μ_{ij} with $1 \leq j \leq l$ to our minimization problem.

II. If one $(\mu_p)_{ij}$ is positive, then from Eq. (160) with this $(\mu_p)_{ij}$ follows that $\frac{1}{n}\sum_{s=1}^{n}(\mu_p)_{sj}\mu_{sj}$ is positive, otherwise Eq. (160) has only negative terms on the left hand side. In particular, the second order necessary conditions are always fulfilled as Eq. (163) is positive. For $(\mu_p)_{ij} < 0$ it follows from Eq. (160) that $\lambda_{ij} > 0$ and from the KKT conditions that $\mu_{ij} = 0$. For $(\mu_p)_{ij} > 0$ it follows from Eq. (160) that $\mu_{ij} > 0$ and from the KKT conditions that $\lambda_{ij} = 0$. Therefore we define:

$$\hat{\mu}_{ij} \;=\; \begin{cases} 0 & \text{for} \quad (\mu_p)_{ij} \;\leq\; 0 \\ (\mu_p)_{ij} & \text{for} \quad (\mu_p)_{ij} \;>\; 0 \end{cases}\,, \tag{165}$$

We write the solution as

$$\mu_{ij} \;=\; \frac{\hat{\mu}_{ij}}{\frac{1}{n}\,\sum_{s=1}^{n}(\mu_p)_{sj}\,\mu_{sj}} \;=\; \alpha_j\,\hat{\mu}_{ij}\,. \tag{166}$$

We now use the equality constraint:

$$\frac{1}{n}\sum_{i=1}^{n}\mu_{ij}^2 \;=\; \alpha_j^2\,\frac{1}{n}\sum_{i=1}^{n}\hat{\mu}_{ij}^2 \;=\; 1\,. \tag{167}$$

Solving for α_j gives:

$$\alpha_j \;=\; \frac{1}{\sqrt{\frac{1}{n}\,\sum_{i=1}^{n}\hat{\mu}_{ij}^2}}\,. \tag{168}$$

Therefore the solution is

$$\mu_{ij} \;=\; \frac{\hat{\mu}_{ij}}{\sqrt{\frac{1}{n}\,\sum_{i=1}^{n}\hat{\mu}_{ij}^2}}\,. \tag{169}$$

This finishes the proof.

□

9.5 Gradient and Scaled Gradient Projection and Projected Newton

9.5.1 Gradient Projection Algorithm

The *projected gradient descent* or *gradient projection algorithm* Bertsekas [1976], Kelley [1999] performs first a gradient step and then projects the result to the *feasible set*. The projection onto the

feasible set is denoted by P, that is, the map that takes $\boldsymbol{\mu}$ into the nearest point (in the L^2-norm) in the feasible set to $\boldsymbol{\mu}$. The feasible set must be convex, however later we will introduce gradient projection methods for non-convex feasible sets.

The gradient projection method is in our case

$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_k \;+\; \lambda\, \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p \;-\; \boldsymbol{\mu}_k)\right) \;. \tag{170}$$

The Lipschitz constant for the gradient is $\|\boldsymbol{\Sigma}_p^{-1}\|_s = e_{\max}(\boldsymbol{\Sigma}_p^{-1})$, the largest eigenvalue of $\boldsymbol{\Sigma}_p^{-1}$. The following statement is Theorem 5.4.5 in Kelley [1999].

Theorem 10 (Theorem 5.4.5 in Kelley [1999])**.** *The* sufficient decrease *condition*

$$D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_{k+1}) \parallel p) \;-\; D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_k) \parallel p) \;\leq\; \frac{-\alpha}{\lambda}\|\boldsymbol{\mu}_k \;-\; \boldsymbol{\mu}_{k+1}\|^2 \tag{171}$$

(e.g. with $\alpha = 10^{-4}$) holds for all λ such that

$$0 \;<\; \lambda \;\leq\; \frac{2\,(1-\alpha)}{e_{\max}(\boldsymbol{\Sigma}_p^{-1})}\,. \tag{172}$$

Proof. See Kelley [1999]. □

Theorem 10 guarantees that we can increase the objective by gradient projection in the E-step, except the case where we already reached the maximum.

For a fast upper bound on the maximal eigenvalue we use

$$e_{\max}(\boldsymbol{\Sigma}_p^{-1}) \;\leq\; \mathrm{Tr}(\boldsymbol{\Sigma}_p^{-1}) \tag{173}$$

and

$$e_{\max}(\boldsymbol{\Sigma}_p^{-1}) \;\leq\; \|\boldsymbol{W}\|_s^2\, \|\boldsymbol{\Psi}^{-1}\|_s \;-1\,, \tag{174}$$

where the latter follows from

$$\boldsymbol{\Sigma}_p^{-1} \;=\; \boldsymbol{I} \;+\; \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\,. \tag{175}$$

Improved methods for finding an appropriate λ by line search methods have been proposed Birgin et al. [2000], Serafini et al. [2005]. We use a search with $\lambda = \beta^t$ with $t = 0, 1, 2, \ldots$ and $\beta = 2^{-1}$ or $\beta = 10^{-1}$.

A special version of the gradient projection method is the *generalized reduced method* Abadie and Carpentier [1969]. This method is able to solve our optimization problem with equality constraints. The gradient projection method has been generalized by Rosen to non-linear constraints Rosen [1961]. The gradient projection algorithm of Rosen can also be used for a region which is not convex. The idea is to linearize the nonlinear constraints and solve the problem. Subsequently a restoration move brings the solution back to the constraint boundaries. Rosen's gradient projection method was improved by Haug and Arora [1979]. *These methods guarantee that we can increase the objective in the E-step for non-convex feasible sets, except the case where we already reached the maximum.* These algorithms for non-convex feasible sets will only give a local maximum. Also the GAM algorithm will only find a local maximum.

9.5.2 Scaled Gradient Projection and Projected Newton Method

Both the *scaled gradient projection algorithm* and the *projected Newton method* were proposed in Bertsekas [1982]. We follow Kelley [1999].

The idea is to use a Newton update instead of the a gradient update:

$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_k \;+\; \lambda\, \boldsymbol{H}^{-1}\, \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p \;-\; \boldsymbol{\mu}_k)\right) \;. \tag{176}$$

$\boldsymbol{H}^{-1}$ can be an arbitrary strict positive definite matrix. If we set $\boldsymbol{H}^{-1} = \boldsymbol{\Sigma}_p$, then we have a Newton update of the *projected Newton method* Bertsekas [1982]. For $\lambda = 1$ we obtain

$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_p\right) \;. \tag{177}$$

otherwise

$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left((1 \;-\; \lambda)\boldsymbol{\mu}_k \;+\; \lambda\boldsymbol{\mu}_p\right) . \tag{178}$$

The search direction for the unconstrained problem can be rotated by $\boldsymbol{H}^{-1}$ to be orthogonal to the direction of decrease in the inactive directions for the constrained problem.

To escape this possible problem, an ϵ-active set is introduced which contains all j with $\mu_j \leq \epsilon$. All columns and rows of the Hessian having an index in the ϵ-active set are fixed to $\boldsymbol{e}_j$. After sorting the indices of the ϵ-active set together, they form a block which is the sub-identity matrix. $\boldsymbol{H}$ is set to the Hessian $\boldsymbol{\Sigma}_p$ where the ϵ-active set columns and rows are replaced by unit vectors.

The following Theorem 11 is Lemma 5.5.1 in Kelley [1999]. *Theorem 11 states that the objective decreases using the reduced Hessian in the projected Newton method for convex feasible sets.*

Theorem 11 (Lemma 5.5.1 in Kelley [1999])**.** *The* sufficient decrease *condition*

$$D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_{k+1}) \parallel p) \;-\; D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_k) \parallel p) \;\leq\; -\,\alpha\,(\boldsymbol{\mu}_k \;-\; \boldsymbol{\mu}_p)^T \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_k \;-\; \boldsymbol{\mu}_{k+1}) \tag{179}$$

holds for all λ smaller than a bound depending on $\boldsymbol{H}$ and ϵ.

Proof. See Kelley [1999]. □

In practical applications, a proper λ is found by line search. The *projected Newton method* uses $\lambda = 1$ to set ϵ Bertsekas [1982]:

$$\epsilon \;=\; \left\|\boldsymbol{\mu}_k \;-\; \mathrm{P}\left(\boldsymbol{\mu}_p\right)\right\| . \tag{180}$$

9.5.3 Combined Method

Following Kim et al. [2006], Serafini et al. [2005] we use the following very general update rule, which includes the gradient projection algorithm, the scaled gradient projection algorithm, and the projected Newton method.

We use following update for the E-step:

$$\begin{aligned} \boldsymbol{d}_{k+1} \;&=\; \mathrm{P}\left(\boldsymbol{\mu}_k \;+\; \lambda\, \boldsymbol{H}^{-1}\, \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p \;-\; \boldsymbol{\mu}_k)\right) , \\ \boldsymbol{\mu}_{k+1} \;&=\; \mathrm{P}\left(\boldsymbol{\mu}_k \;+\; \gamma\,(\boldsymbol{d}_{k+1} \;-\; \boldsymbol{\mu}_k)\right) . \end{aligned} \tag{181}$$

We have to project twice since the equality constraint produces a manifold in the parameter space.

We iterate this update until we see a decrease of the objective in the E-step:

$$D_{\mathrm{KL}}(Q_{k+1} \parallel p) \;-\; D_{\mathrm{KL}}(Q_k \parallel p) \;<\; 0\,. \tag{182}$$

For the constraints we have only to optimize the mean vector $\boldsymbol{\mu}$ to ensure

$$D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_{k+1}) \parallel p) \;-\; D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_k) \parallel p) \;<\; 0\,. \tag{183}$$

Even

$$D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_{k+1}) \parallel p) \;=\; D_{\mathrm{KL}}(Q(\boldsymbol{\mu}_k) \parallel p) \tag{184}$$

can be sufficient if minimizing $\boldsymbol{\Sigma}_{k+1} = \boldsymbol{\Sigma}_p$ ensures

$$D_{\mathrm{KL}}(Q_{k+1} \parallel p) \;<\; D_{\mathrm{KL}}(Q_k \parallel p)\,. \tag{185}$$

We use following schedule:

1.
   - $\boldsymbol{H}^{-1} = \boldsymbol{\Sigma}_p$
   - $\lambda = 1$
   - $\gamma = 1$

   That is

$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left(\boldsymbol{\mu}_p\right) . \tag{186}$$

2. • $\boldsymbol{H}^{-1} = \boldsymbol{\Sigma}_p$
   • $\lambda = 1$
   • $\gamma \in (0, 1]$
   That is
$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left((1 \;-\; \gamma)\, \boldsymbol{\mu}_k \;+\; \gamma\, \mathrm{P}\left(\boldsymbol{\mu}_p\right)\right) \;. \tag{187}$$

3. • $\boldsymbol{H}^{-1} = \boldsymbol{\Sigma}_p$
   • $\lambda \in (0, 1]$
   • $\gamma = 1$
   That is
$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left((1 \;-\; \lambda)\boldsymbol{\mu}_k \;+\; \lambda\boldsymbol{\mu}_p\right) \;. \tag{188}$$

4. • $\boldsymbol{H}^{-1} = \boldsymbol{\Sigma}_p$
   • $\lambda \in (0, 1]$
   • $\gamma =\in (0, 1]$
   That is
$$\boldsymbol{\mu}_{k+1} \;=\; \mathrm{P}\left((1 \;-\; \gamma)\, \boldsymbol{\mu}_k \;+\; \gamma\, \mathrm{P}\left((1 \;-\; \lambda)\boldsymbol{\mu}_k \;+\; \lambda\boldsymbol{\mu}_p\right)\right) \;. \tag{189}$$

5. • $\boldsymbol{H}^{-1} = \mathrm{R}(\boldsymbol{\Sigma}_p)$
   • $\lambda \in (0, 1]$
   • $\gamma =\in (0, 1]$
   $\mathrm{R}(\boldsymbol{\Sigma}_p)$ denotes the reduced matrix (Hessian or a positive definite) according to the projected Newton method or the scaled gradient projection algorithm. For convex feasible sets we can guarantee at this level already an increase of the objective at the E-step.

6. • $\boldsymbol{H}^{-1} = \boldsymbol{I}$
   • $\lambda \in (0, 1]$
   • $\gamma =\in (0, 1]$
   This is the gradient projection algorithm. In particular we include the generalized reduced method and Rosen's gradient projection method. At this step we guarantee an increase of the objective at the E-step even for non-convex feasible sets because we also use complex methods for constraint optimization.

Step 5. ensures an improvement if only using rectifying constraints according to the theory of projected Newton methods Kelley [1999]. Step 6. ensures an improvement if using both rectifying constraints and normalizing constraints, because we use known methods for constraint optimization. To set $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k$ is sufficient to increase the objective at the E-step if $\boldsymbol{\Sigma}_{k+1} = \boldsymbol{\Sigma}_p$ decreases the KL divergence. However we will not always set $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k$ to avoid accumulation points outside the solution set.

10 Alternative Gaussian Prior

We assume $\boldsymbol{h}$ is Gaussian with covariance $\boldsymbol{M}$ and mean $\boldsymbol{\xi}$

$$\boldsymbol{h} \;\sim\; \mathcal{N}\left(\boldsymbol{\xi}, \boldsymbol{M}\right) \;. \tag{190}$$

We derive the posterior for this prior.

The likelihood is Gaussian since a affine transformation of a Gaussian random variable is again a Gaussian random variable and the convolution of two Gaussians is Gaussian, too. Thus, $\boldsymbol{v} = \boldsymbol{W}\boldsymbol{h} + \boldsymbol{\epsilon}$ is Gaussian if $\boldsymbol{h}$ and $\boldsymbol{\epsilon}$ are both Gaussian. For the prior moments we have

$$\mathrm{E}(\boldsymbol{h}) \;=\; \boldsymbol{\xi}\,, \tag{191}$$
$$\mathrm{E}(\boldsymbol{h}\boldsymbol{h}^T) \;=\; \boldsymbol{M} \;+\; \boldsymbol{\xi}\,\boldsymbol{\xi}^T\,, \tag{192}$$
$$\mathrm{var}(\boldsymbol{h}) \;=\; \boldsymbol{M} \tag{193}$$

and for the likelihood of $\boldsymbol{v}$ we obtain the moments

$$\begin{aligned} \mathrm{E}(\boldsymbol{v}) &= \boldsymbol{W}\boldsymbol{\xi}\,, && (194)\\ \mathrm{E}(\boldsymbol{v}\boldsymbol{v}^T) &= \boldsymbol{W}\,\mathrm{E}(\boldsymbol{h}\boldsymbol{h}^T)\,\boldsymbol{W}^T + \boldsymbol{\Psi} && (195)\\ &= \boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi} + \boldsymbol{W}\,\boldsymbol{\xi}\,\boldsymbol{\xi}^T\,\boldsymbol{W}^T\,, \\ \mathrm{var}(\boldsymbol{v}) &= \boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\,. && (196) \end{aligned}$$

We need some algebraic identities to derive the posterior. The Woodbury matrix identity gives

$$\boldsymbol{M} - \boldsymbol{M}\,\boldsymbol{W}^T\,\left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W}\,\boldsymbol{M} = \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,. \tag{197}$$

Multiplying this equation from the left hand side with $\boldsymbol{\Psi}^{-1}\boldsymbol{W}$ gives

$$\begin{aligned} &\boldsymbol{\Psi}^{-1}\,\boldsymbol{W}\,\left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1} && (198)\\ &= \boldsymbol{\Psi}^{-1}\,\boldsymbol{W}\,\boldsymbol{M} - \boldsymbol{\Psi}^{-1}\,\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T\,\left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W}\,\boldsymbol{M} \\ &= \boldsymbol{\Psi}^{-1}\left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)\left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W}\,\boldsymbol{M} - \\ &\quad \boldsymbol{\Psi}^{-1}\,\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T\,\left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W}\,\boldsymbol{M} \\ &= \left(\boldsymbol{\Psi}^{-1}\left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right) - \boldsymbol{\Psi}^{-1}\,\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T\right)\left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W}\,\boldsymbol{M} \\ &= \left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{W}\,\boldsymbol{M}\,. \end{aligned}$$

It follows that

$$\boldsymbol{M}\,\boldsymbol{W}^T\,\left(\boldsymbol{W}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{a} = \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\,\boldsymbol{a}\,. \tag{199}$$

The posterior $p(\boldsymbol{h} \mid \boldsymbol{v})$ is derived from Gaussian conditioning because both the likelihood $p(\boldsymbol{v})$ and the prior $p(\boldsymbol{h})$ are Gaussian distributed. The conditional distribution $p(\boldsymbol{a} \mid \boldsymbol{b})$ of two random variables $\boldsymbol{a}$ and $\boldsymbol{b}$ that both follow a Gaussian distribution is a Gaussian:

$$\begin{aligned} \boldsymbol{a} &\sim \mathcal{N}\left(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}\right)\,, && (200)\\ \boldsymbol{b} &\sim \mathcal{N}\left(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb}\right)\,, && (201)\\ \boldsymbol{\Sigma}_{ba} &= \mathrm{Cov}(\boldsymbol{b}, \boldsymbol{a})\,, && (202)\\ \boldsymbol{\Sigma}_{ab} &= \mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b})\,, && (203)\\ \boldsymbol{a} \mid \boldsymbol{b} &\sim \mathcal{N}\left(\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\left(\boldsymbol{b} - \boldsymbol{\mu}_b\right)\,,\, \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\right)\,. && (204) \end{aligned}$$

Therefore we need the second moments between $\boldsymbol{v}$ and $\boldsymbol{h}$:

$$\mathrm{E}(\boldsymbol{v}\boldsymbol{h}^T) = \mathrm{E}(\boldsymbol{W}\boldsymbol{h}\boldsymbol{h}^T) + \mathrm{E}(\boldsymbol{\epsilon}\boldsymbol{h}^T) = \boldsymbol{W}\,\left(\boldsymbol{M} + \boldsymbol{\xi}\,\boldsymbol{\xi}^T\right)\,. \tag{205}$$

The covariances between $\boldsymbol{v}$ and $\boldsymbol{h}$ are

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{v}, \boldsymbol{h}) &= \mathrm{E}(\boldsymbol{v}\boldsymbol{h}^T) - \mathrm{E}(\boldsymbol{v})\mathrm{E}(\boldsymbol{h}^T) && (206)\\ &= \boldsymbol{W}\,\boldsymbol{M} + \boldsymbol{W}\boldsymbol{\xi}\,\boldsymbol{\xi}^T - \boldsymbol{W}\boldsymbol{\xi}\,\boldsymbol{\xi}^T = \boldsymbol{W}\,\boldsymbol{M}\,, \\ \mathrm{Cov}(\boldsymbol{h}, \boldsymbol{v}) &= \mathrm{E}(\boldsymbol{h}\boldsymbol{v}^T) - \mathrm{E}(\boldsymbol{h})\mathrm{E}(\boldsymbol{v}^T) = \boldsymbol{M}\,\boldsymbol{W}^T\,. && (207) \end{aligned}$$

Thus, the mean of $p(\boldsymbol{h} \mid \boldsymbol{v})$ is

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{h}|\boldsymbol{v}} &= \boldsymbol{\xi} + \boldsymbol{M}\,\boldsymbol{W}^T\,(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi})^{-1}(\boldsymbol{v} - \boldsymbol{W}\boldsymbol{\xi}) && (208)\\ &= \boldsymbol{\xi} + \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\,(\boldsymbol{v} + \boldsymbol{W}\boldsymbol{\xi}) \\ &= \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,\left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)\boldsymbol{\xi} \\ &\quad + \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\,\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\,(\boldsymbol{v} - \boldsymbol{W}\boldsymbol{\xi}) \\ &= \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1} \\ &\quad \left(\boldsymbol{M}^{-1}\,\boldsymbol{\xi} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\,\boldsymbol{\xi} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\,\boldsymbol{v} - \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\boldsymbol{\xi}\right) \\ &= \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}\left(\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\,\boldsymbol{v} + \boldsymbol{M}^{-1}\,\boldsymbol{\xi}\right)\,. \end{aligned}$$

The covariance matrix of $p(\boldsymbol{h} \mid \boldsymbol{v})$ is

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{h}|\boldsymbol{v}} &= \boldsymbol{M} - \boldsymbol{M}\,\boldsymbol{W}^T \left(\boldsymbol{W}\,\boldsymbol{M}\,\boldsymbol{W}^T + \boldsymbol{\Psi}\right)^{-1} \boldsymbol{W}\,\boldsymbol{M} \\ &= \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} . \end{aligned} \tag{209}$$

In particular, the variable $\boldsymbol{\xi}$ may be used to enforce more sparseness by setting its components to negative values. Since the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{h}|\boldsymbol{v}}$ is positive semi-definite, we ensure that

$$\boldsymbol{\xi}^T \left(\boldsymbol{M}^{-1} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W}\right)^{-1} \boldsymbol{\xi} \geq 0 . \tag{210}$$

If $\boldsymbol{\xi} = -\rho \mathbf{1}$ ($\mathbf{1}$ is the vector with all components being one), then the largest absolute components of $\boldsymbol{\Sigma}_{\boldsymbol{h}|\boldsymbol{v}} \boldsymbol{\xi}$ must be negative. Thus, $\boldsymbol{\xi} = -\rho \mathbf{1}$ leads to sparser solutions.

11 Hyperparameters Selected for Method Assessment

The performance of rectified factor networks (RFNs) as unsupervised methods for data representation was compared with:
(1) **RFN**: rectified factor networks,
(2) **RFNn**: RFNs without normalization,
(3) **DAE**: denoising autoencoders with rectified linear units,
(4) **RBM**: restricted Boltzmann machines with Gaussian visible units and hidden binary units,
(5) **FAsp**: factor analysis with Jeffrey's prior ($p(z) \propto 1/z$) on the hidden units which is sparser than a Laplace prior,
(6) **FAlap**: factor analysis with Laplace prior on the hidden units,
(7) **ICA**: independent component analysis by FastICA Hyvärinen and Oja [1999],
(8) **SFA**: sparse factor analysis with a Laplace prior on the parameters,
(9) **FA**: standard factor analysis,
(10) **PCA**: principal component analysis.
The number of components are fixed to 50, 100, or 150 for each method. The used hyperparameters are listed in Tab. 1.

Table 1: Hyperparameters of all methods that were used to assess the performance of rectified factor networks (RFNs) as unsupervised methods for data representation.

Method	**Used hyperparameters**
RFN	{learning rate=0.1, iterations=1000}
RFNn	{learning rate=0.1, iterations=1000}
DAE	{corruption level=0.2, learning rate=1e-04, iterations=1000}
RBM	{learning rate=0.01, iterations=1000}
FAsp	{iterations=500}
FAlap	{iterations=500}
SFA	{Laplace weight decay factor=5e-05, iterations=500}

12 Data Set I

The number of components are fixed to 50, 100 or 150.

We generated nine different benchmark data sets (D1 to D9), where each data set consists of 100 instances for averaging the results. Each instance consists of 100 samples and 100 features resulting in a 100×100 data matrix. Into these data matrices, structures are implanted as biclusters **?**. A bicluster is a pattern consisting of a particular number of features which is found in a particular number of samples. The size of the bicluster is given by the number of features that form the pattern and by the number of samples in which the pattern is found. The data sets had different noise levels and different bicluster sizes. We considered large and small bicluster sizes, where large biclusters have 20–30 samples and 20–30 features, while small biclusters have 3–8 samples and 3–8 features. The signal strength (scaling factor) of a pattern in a sample was randomly chosen according to the Gaussian $\mathcal{N}(1, 1)$. Finally, to each data matrix background noise was added, where the noise is distributed according to a zero-mean Gaussian with standard deviation 1, 5, or 10. The data sets are described in Tab. 2. The remaining components of the spanning outer product vectors were drawn by $\mathcal{N}(0, 0.01)$.

Table 2: Overview over the datasets. Shown is the background noise ("noise"), the number of large biclusters (n_1), and the number of small biclusters (n_2).

	D1	D2	D3	D4	D5	D6	D7	D8	D9
noise	1	5	10	1	5	10	1	5	10
n_1	10	10	10	15	15	15	5	5	5
n_2	10	10	10	5	5	5	15	15	15

Table 3: Comparison for 50 factors / hidden units extracted by RFN, RFN without normalization (RFNn), denoising autoencoder (DAE), restricted Boltzmann machines (RBM), factor analysis with a very sparse prior (FAsp), factor analysis with a Laplace prior (FAlap), independent component analysis (ICA), sparse factor analysis (SFA), factor analysis (FA), and principal component analysis (PCA) on nine data sets. Criteria are: sparseness of the coding units (SP), reconstruction error (ER), and the difference between the empirical and the model covariance matrix (CO). The lower right column block gives the average SP (%), ER and CO. Results reported here, are the mean together with the standard deviation of 100 instances. The maximal value in the table and the maximal standard deviation was set to 999 and to 99, respectively.

	D1			D2			D3			D4			D5		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$74_{\pm 0}$	$58_{\pm 1}$	$5_{\pm 0}$	$75_{\pm 0}$	$233_{\pm 3}$	$66_{\pm 1}$	$75_{\pm 0}$	$456_{\pm 5}$	$253_{\pm 6}$	$74_{\pm 0}$	$63_{\pm 1}$	$6_{\pm 1}$	$75_{\pm 0}$	$236_{\pm 3}$	$68_{\pm 2}$
RFNn	$73_{\pm 0}$	$85_{\pm 3}$	$13_{\pm 2}$	$75_{\pm 0}$	$272_{\pm 3}$	$85_{\pm 2}$	$75_{\pm 0}$	$531_{\pm 6}$	$321_{\pm 7}$	$72_{\pm 0}$	$95_{\pm 4}$	$17_{\pm 2}$	$74_{\pm 0}$	$276_{\pm 4}$	$89_{\pm 3}$
DAE	$65_{\pm 0}$	$65_{\pm 2}$	—	$66_{\pm 0}$	$233_{\pm 2}$	—	$66_{\pm 0}$	$456_{\pm 4}$	—	$65_{\pm 1}$	$71_{\pm 2}$	—	$66_{\pm 0}$	$237_{\pm 2}$	—
RBM	$25_{\pm 2}$	$86_{\pm 3}$	—	$11_{\pm 1}$	$287_{\pm 3}$	—	$10_{\pm 1}$	$558_{\pm 5}$	—	$25_{\pm 2}$	$94_{\pm 3}$	—	$11_{\pm 1}$	$292_{\pm 3}$	—
FAsp	$39_{\pm 1}$	$232_{\pm 31}$	$654_{\pm 99}$	$40_{\pm 1}$	$999_{\pm 41}$	$999_{\pm 99}$	$41_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$38_{\pm 1}$	$318_{\pm 33}$	$999_{\pm 99}$	$40_{\pm 1}$	$999_{\pm 48}$	$999_{\pm 99}$
FAlap	$4_{\pm 0}$	$53_{\pm 2}$	$144_{\pm 36}$	$4_{\pm 0}$	$224_{\pm 5}$	$185_{\pm 5}$	$5_{\pm 0}$	$439_{\pm 9}$	$692_{\pm 16}$	$4_{\pm 0}$	$55_{\pm 2}$	$180_{\pm 39}$	$4_{\pm 0}$	$226_{\pm 5}$	$192_{\pm 6}$
ICA	$2_{\pm 0}$	$34_{\pm 0}$	—	$2_{\pm 0}$	$164_{\pm 2}$	—	$2_{\pm 0}$	$324_{\pm 4}$	—	$2_{\pm 0}$	$35_{\pm 0}$	—	$2_{\pm 0}$	$166_{\pm 2}$	—
SFA	$1_{\pm 0}$	$42_{\pm 1}$	$11_{\pm 2}$	$1_{\pm 0}$	$206_{\pm 4}$	$56_{\pm 2}$	$1_{\pm 0}$	$406_{\pm 9}$	$215_{\pm 7}$	$1_{\pm 0}$	$42_{\pm 1}$	$13_{\pm 2}$	$1_{\pm 0}$	$208_{\pm 4}$	$58_{\pm 2}$
FA	$1_{\pm 0}$	$42_{\pm 1}$	$6_{\pm 1}$	$1_{\pm 0}$	$206_{\pm 4}$	$54_{\pm 2}$	$1_{\pm 0}$	$407_{\pm 8}$	$210_{\pm 6}$	$1_{\pm 0}$	$42_{\pm 1}$	$8_{\pm 1}$	$1_{\pm 0}$	$208_{\pm 4}$	$56_{\pm 2}$
PCA	$1_{\pm 0}$	$34_{\pm 0}$	—	$0_{\pm 0}$	$164_{\pm 2}$	—	$0_{\pm 0}$	$324_{\pm 4}$	—	$1_{\pm 0}$	$35_{\pm 0}$	—	$0_{\pm 0}$	$166_{\pm 2}$	—

	D6			D7			D8			D9			average		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$75_{\pm 0}$	$458_{\pm 5}$	$256_{\pm 6}$	$75_{\pm 0}$	$53_{\pm 1}$	$4_{\pm 1}$	$75_{\pm 0}$	$230_{\pm 3}$	$64_{\pm 1}$	$75_{\pm 0}$	$454_{\pm 5}$	$251_{\pm 5}$	$75_{\pm 0}$	$249_{\pm 3}$	$108_{\pm 3}$
RFNn	$75_{\pm 0}$	$532_{\pm 6}$	$323_{\pm 7}$	$73_{\pm 0}$	$73_{\pm 3}$	$10_{\pm 2}$	$75_{\pm 0}$	$268_{\pm 3}$	$82_{\pm 2}$	$75_{\pm 0}$	$528_{\pm 6}$	$317_{\pm 7}$	$74_{\pm 0}$	$295_{\pm 4}$	$140_{\pm 4}$
DAE	$66_{\pm 0}$	$458_{\pm 4}$	—	$65_{\pm 0}$	$58_{\pm 1}$	—	$66_{\pm 0}$	$230_{\pm 2}$	—	$66_{\pm 0}$	$453_{\pm 5}$	—	$66_{\pm 0}$	$251_{\pm 3}$	—
RBM	$10_{\pm 1}$	$561_{\pm 5}$	—	$23_{\pm 2}$	$76_{\pm 2}$	—	$11_{\pm 1}$	$282_{\pm 3}$	—	$10_{\pm 1}$	$555_{\pm 5}$	—	$15_{\pm 1}$	$310_{\pm 4}$	—
FAsp	$40_{\pm 2}$	$999_{\pm 99}$	$999_{\pm 99}$	$39_{\pm 1}$	$152_{\pm 26}$	$345_{\pm 99}$	$40_{\pm 1}$	$999_{\pm 31}$	$999_{\pm 99}$	$41_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$40_{\pm 1}$	$999_{\pm 63}$	$999_{\pm 99}$
FAlap	$5_{\pm 0}$	$443_{\pm 9}$	$701_{\pm 15}$	$4_{\pm 0}$	$50_{\pm 2}$	$110_{\pm 37}$	$4_{\pm 0}$	$221_{\pm 5}$	$177_{\pm 4}$	$5_{\pm 0}$	$439_{\pm 10}$	$686_{\pm 15}$	$4_{\pm 0}$	$239_{\pm 6}$	$341_{\pm 19}$
ICA	$2_{\pm 0}$	$325_{\pm 4}$	—	$2_{\pm 0}$	$34_{\pm 0}$	—	$2_{\pm 0}$	$163_{\pm 2}$	—	$2_{\pm 0}$	$322_{\pm 4}$	—	$2_{\pm 0}$	$174_{\pm 2}$	—
SFA	$1_{\pm 0}$	$408_{\pm 9}$	$217_{\pm 7}$	$1_{\pm 0}$	$42_{\pm 1}$	$8_{\pm 2}$	$1_{\pm 0}$	$204_{\pm 4}$	$54_{\pm 2}$	$1_{\pm 0}$	$405_{\pm 9}$	$213_{\pm 7}$	$1_{\pm 0}$	$218_{\pm 5}$	$94_{\pm 3}$
FA	$1_{\pm 0}$	$409_{\pm 9}$	$212_{\pm 7}$	$1_{\pm 0}$	$42_{\pm 1}$	$4_{\pm 1}$	$1_{\pm 0}$	$205_{\pm 4}$	$53_{\pm 2}$	$1_{\pm 0}$	$405_{\pm 8}$	$208_{\pm 6}$	$1_{\pm 0}$	$218_{\pm 4}$	$90_{\pm 3}$
PCA	$0_{\pm 0}$	$325_{\pm 4}$	—	$1_{\pm 0}$	$34_{\pm 0}$	—	$0_{\pm 0}$	$163_{\pm 2}$	—	$0_{\pm 0}$	$322_{\pm 4}$	—	$0_{\pm 0}$	$174_{\pm 2}$	—

Table 4: Comparison for 100 factors / hidden units extracted by RFN, RFN without normalization (RFNn), denoising autoencoder (DAE), restricted Boltzmann machines (RBM), factor analysis with a very sparse prior (FAsp), factor analysis with a Laplace prior (FAlap), independent component analysis (ICA), sparse factor analysis (SFA), factor analysis (FA), and principal component analysis (PCA) on nine data sets. Criteria are: sparseness of the coding units (SP), reconstruction error (ER), and the difference between the empirical and the model covariance matrix (CO). The lower right column block gives the average SP (%), ER and CO. Results reported here, are the mean together with the standard deviation of 100 instances. The maximal value in the table and the maximal standard deviation was set to 999 and to 99, respectively.

	D1			D2			D3			D4			D5		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$79_{\pm 1}$	$23_{\pm 3}$	$2_{\pm 0}$	$82_{\pm 1}$	$63_{\pm 9}$	$16_{\pm 3}$	$82_{\pm 1}$	$120_{\pm 17}$	$61_{\pm 15}$	$78_{\pm 1}$	$27_{\pm 3}$	$2_{\pm 1}$	$82_{\pm 1}$	$62_{\pm 7}$	$16_{\pm 3}$
RFNn	$77_{\pm 0}$	$61_{\pm 4}$	$6_{\pm 1}$	$80_{\pm 0}$	$169_{\pm 4}$	$36_{\pm 2}$	$80_{\pm 0}$	$326_{\pm 8}$	$135_{\pm 6}$	$76_{\pm 1}$	$73_{\pm 4}$	$9_{\pm 2}$	$79_{\pm 0}$	$171_{\pm 5}$	$37_{\pm 2}$
DAE	$67_{\pm 0}$	$48_{\pm 2}$	—	$70_{\pm 0}$	$134_{\pm 1}$	—	$70_{\pm 0}$	$260_{\pm 2}$	—	$67_{\pm 0}$	$54_{\pm 2}$	—	$70_{\pm 0}$	$137_{\pm 1}$	—
RBM	$14_{\pm 1}$	$81_{\pm 3}$	—	$4_{\pm 0}$	$266_{\pm 3}$	—	$4_{\pm 0}$	$514_{\pm 6}$	—	$15_{\pm 1}$	$88_{\pm 2}$	—	$4_{\pm 0}$	$270_{\pm 3}$	—
FAsp	$72_{\pm 0}$	$233_{\pm 32}$	$499_{\pm 99}$	$62_{\pm 0}$	$999_{\pm 43}$	$999_{\pm 99}$	$56_{\pm 0}$	$999_{\pm 99}$	$999_{\pm 99}$	$71_{\pm 0}$	$320_{\pm 34}$	$878_{\pm 99}$	$62_{\pm 0}$	$999_{\pm 49}$	$999_{\pm 99}$
FAlap	$6_{\pm 0}$	$27_{\pm 3}$	$202_{\pm 17}$	$6_{\pm 0}$	$38_{\pm 3}$	$756_{\pm 33}$	$6_{\pm 0}$	$74_{\pm 5}$	$999_{\pm 83}$	$6_{\pm 0}$	$31_{\pm 3}$	$274_{\pm 23}$	$6_{\pm 0}$	$39_{\pm 3}$	$778_{\pm 34}$
ICA	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$6_{\pm 0}$	$30_{\pm 5}$	$1_{\pm 0}$	$14_{\pm 0}$	$68_{\pm 3}$	$1_{\pm 0}$	$28_{\pm 1}$	$243_{\pm 8}$	$1_{\pm 0}$	$8_{\pm 0}$	$38_{\pm 5}$	$1_{\pm 0}$	$15_{\pm 0}$	$72_{\pm 3}$
FA	$1_{\pm 0}$	$6_{\pm 0}$	$18_{\pm 3}$	$1_{\pm 0}$	$14_{\pm 0}$	$50_{\pm 2}$	$1_{\pm 0}$	$28_{\pm 1}$	$182_{\pm 7}$	$1_{\pm 0}$	$8_{\pm 0}$	$24_{\pm 4}$	$1_{\pm 0}$	$15_{\pm 0}$	$52_{\pm 2}$
PCA	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

	D6			D7			D8			D9			average		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$82_{\pm 1}$	$120_{\pm 16}$	$60_{\pm 13}$	$80_{\pm 1}$	$18_{\pm 2}$	$1_{\pm 0}$	$82_{\pm 1}$	$61_{\pm 7}$	$15_{\pm 3}$	$82_{\pm 1}$	$122_{\pm 13}$	$60_{\pm 11}$	$81_{\pm 1}$	$68_{\pm 9}$	$26_{\pm 6}$
RFNn	$80_{\pm 0}$	$329_{\pm 7}$	$137_{\pm 6}$	$78_{\pm 0}$	$49_{\pm 3}$	$4_{\pm 1}$	$80_{\pm 0}$	$165_{\pm 4}$	$34_{\pm 1}$	$80_{\pm 0}$	$325_{\pm 7}$	$134_{\pm 6}$	$79_{\pm 0}$	$185_{\pm 5}$	$59_{\pm 3}$
DAE	$70_{\pm 0}$	$261_{\pm 2}$	—	$68_{\pm 0}$	$39_{\pm 2}$	—	$70_{\pm 0}$	$132_{\pm 1}$	—	$70_{\pm 0}$	$259_{\pm 2}$	—	$69_{\pm 0}$	$147_{\pm 2}$	—
RBM	$4_{\pm 0}$	$517_{\pm 6}$	—	$12_{\pm 1}$	$71_{\pm 2}$	—	$4_{\pm 0}$	$261_{\pm 3}$	—	$4_{\pm 0}$	$512_{\pm 5}$	—	$7_{\pm 1}$	$287_{\pm 4}$	—
FAsp	$56_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$73_{\pm 0}$	$149_{\pm 28}$	$237_{\pm 62}$	$62_{\pm 0}$	$999_{\pm 34}$	$999_{\pm 99}$	$56_{\pm 0}$	$999_{\pm 99}$	$999_{\pm 99}$	$63_{\pm 0}$	$999_{\pm 65}$	$999_{\pm 99}$
FAlap	$6_{\pm 0}$	$74_{\pm 6}$	$999_{\pm 91}$	$6_{\pm 0}$	$22_{\pm 3}$	$134_{\pm 14}$	$6_{\pm 0}$	$37_{\pm 2}$	$733_{\pm 28}$	$6_{\pm 0}$	$73_{\pm 6}$	$999_{\pm 84}$	$6_{\pm 0}$	$46_{\pm 4}$	$985_{\pm 45}$
ICA	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$28_{\pm 1}$	$247_{\pm 8}$	$1_{\pm 0}$	$5_{\pm 0}$	$21_{\pm 5}$	$1_{\pm 0}$	$14_{\pm 0}$	$64_{\pm 2}$	$1_{\pm 0}$	$27_{\pm 1}$	$240_{\pm 7}$	$1_{\pm 0}$	$16_{\pm 1}$	$114_{\pm 5}$
FA	$1_{\pm 0}$	$28_{\pm 1}$	$184_{\pm 8}$	$1_{\pm 0}$	$5_{\pm 0}$	$11_{\pm 3}$	$1_{\pm 0}$	$14_{\pm 0}$	$47_{\pm 2}$	$1_{\pm 0}$	$27_{\pm 1}$	$179_{\pm 7}$	$1_{\pm 0}$	$16_{\pm 1}$	$83_{\pm 4}$
PCA	$1_{\pm 0}$	$0_{\pm 0}$	—	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

Table 5: Comparison for 150 factors / hidden units extracted by RFN, RFN without normalization (RFNn), denoising autoencoder (DAE), restricted Boltzmann machines (RBM), factor analysis with a very sparse prior (FAsp), factor analysis with a Laplace prior (FAlap), independent component analysis (ICA), sparse factor analysis (SFA), factor analysis (FA), and principal component analysis (PCA) on nine data sets. Criteria are: sparseness of the coding units (SP), reconstruction error (ER), and the difference between the empirical and the model covariance matrix (CO). The lower right column block gives the average SP (%), ER and CO. The lower right column block gives the average SP, ER and CO. Results reported here, are the mean together with the standard deviation of 100 instances. The maximal value in the table and the maximal standard deviation was set to 999 and to 99, respectively.

	D1			D2			D3			D4			D5		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$83_{\pm 1}$	$7_{\pm 2}$	$0_{\pm 1}$	$86_{\pm 0}$	$15_{\pm 1}$	$3_{\pm 1}$	$86_{\pm 2}$	$33_{\pm 20}$	$18_{\pm 23}$	$83_{\pm 1}$	$9_{\pm 2}$	$1_{\pm 0}$	$86_{\pm 1}$	$15_{\pm 3}$	$4_{\pm 1}$
RFNn	$79_{\pm 0}$	$48_{\pm 3}$	$4_{\pm 1}$	$81_{\pm 0}$	$129_{\pm 3}$	$21_{\pm 1}$	$81_{\pm 0}$	$250_{\pm 7}$	$80_{\pm 4}$	$78_{\pm 0}$	$60_{\pm 4}$	$6_{\pm 1}$	$81_{\pm 0}$	$131_{\pm 3}$	$22_{\pm 1}$
DAE	$68_{\pm 0}$	$44_{\pm 2}$	—	$72_{\pm 0}$	$118_{\pm 1}$	—	$72_{\pm 0}$	$229_{\pm 2}$	—	$68_{\pm 0}$	$50_{\pm 2}$	—	$72_{\pm 0}$	$120_{\pm 2}$	—
RBM	$10_{\pm 1}$	$81_{\pm 3}$	—	$3_{\pm 0}$	$265_{\pm 3}$	—	$3_{\pm 0}$	$514_{\pm 6}$	—	$10_{\pm 1}$	$88_{\pm 2}$	—	$3_{\pm 0}$	$270_{\pm 4}$	—
FAsp	$83_{\pm 1}$	$233_{\pm 32}$	$340_{\pm 71}$	$79_{\pm 0}$	$999_{\pm 43}$	$999_{\pm 99}$	$77_{\pm 0}$	$999_{\pm 99}$	$999_{\pm 99}$	$81_{\pm 1}$	$320_{\pm 34}$	$574_{\pm 99}$	$79_{\pm 1}$	$999_{\pm 49}$	$999_{\pm 99}$
FAlap	$4_{\pm 0}$	$27_{\pm 3}$	$295_{\pm 25}$	$4_{\pm 0}$	$38_{\pm 3}$	$791_{\pm 41}$	$3_{\pm 0}$	$74_{\pm 5}$	$999_{\pm 91}$	$4_{\pm 0}$	$31_{\pm 3}$	$394_{\pm 31}$	$4_{\pm 0}$	$39_{\pm 3}$	$817_{\pm 39}$
ICA	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$6_{\pm 0}$	$49_{\pm 7}$	$1_{\pm 0}$	$14_{\pm 0}$	$173_{\pm 4}$	$1_{\pm 0}$	$28_{\pm 1}$	$632_{\pm 10}$	$1_{\pm 0}$	$8_{\pm 0}$	$61_{\pm 7}$	$1_{\pm 0}$	$15_{\pm 0}$	$181_{\pm 5}$
FA	$1_{\pm 0}$	$6_{\pm 0}$	$40_{\pm 5}$	$1_{\pm 0}$	$14_{\pm 0}$	$160_{\pm 4}$	$1_{\pm 0}$	$28_{\pm 1}$	$590_{\pm 10}$	$1_{\pm 0}$	$8_{\pm 0}$	$51_{\pm 6}$	$1_{\pm 0}$	$15_{\pm 0}$	$168_{\pm 4}$
PCA	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

	D6			D7			D8			D9			average		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$86_{\pm 1}$	$30_{\pm 13}$	$15_{\pm 16}$	$84_{\pm 2}$	$5_{\pm 3}$	$0_{\pm 1}$	$86_{\pm 0}$	$14_{\pm 1}$	$3_{\pm 1}$	$86_{\pm 1}$	$30_{\pm 8}$	$15_{\pm 9}$	$85_{\pm 1}$	$17_{\pm 6}$	$7_{\pm 6}$
RFNn	$81_{\pm 0}$	$251_{\pm 6}$	$81_{\pm 3}$	$80_{\pm 0}$	$37_{\pm 3}$	$2_{\pm 0}$	$81_{\pm 0}$	$126_{\pm 3}$	$20_{\pm 1}$	$81_{\pm 0}$	$248_{\pm 6}$	$79_{\pm 3}$	$80_{\pm 0}$	$142_{\pm 4}$	$35_{\pm 2}$
DAE	$72_{\pm 0}$	$230_{\pm 2}$	—	$70_{\pm 0}$	$36_{\pm 2}$	—	$72_{\pm 0}$	$116_{\pm 1}$	—	$72_{\pm 0}$	$227_{\pm 2}$	—	$71_{\pm 0}$	$130_{\pm 2}$	—
RBM	$3_{\pm 0}$	$516_{\pm 6}$	—	$8_{\pm 1}$	$71_{\pm 2}$	—	$3_{\pm 0}$	$260_{\pm 4}$	—	$3_{\pm 0}$	$511_{\pm 5}$	—	$5_{\pm 0}$	$286_{\pm 4}$	—
FAsp	$77_{\pm 0}$	$999_{\pm 99}$	$999_{\pm 99}$	$84_{\pm 0}$	$149_{\pm 28}$	$168_{\pm 55}$	$80_{\pm 0}$	$999_{\pm 34}$	$999_{\pm 99}$	$77_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$80_{\pm 0}$	$999_{\pm 65}$	$999_{\pm 99}$
FAlap	$3_{\pm 0}$	$74_{\pm 6}$	$999_{\pm 97}$	$4_{\pm 0}$	$22_{\pm 3}$	$198_{\pm 17}$	$4_{\pm 0}$	$37_{\pm 2}$	$768_{\pm 40}$	$3_{\pm 0}$	$73_{\pm 6}$	$999_{\pm 93}$	$4_{\pm 0}$	$46_{\pm 4}$	$976_{\pm 53}$
ICA	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$28_{\pm 1}$	$640_{\pm 11}$	$1_{\pm 0}$	$5_{\pm 0}$	$34_{\pm 6}$	$1_{\pm 0}$	$14_{\pm 0}$	$164_{\pm 3}$	$1_{\pm 0}$	$27_{\pm 1}$	$625_{\pm 9}$	$1_{\pm 0}$	$16_{\pm 1}$	$285_{\pm 7}$
FA	$1_{\pm 0}$	$28_{\pm 1}$	$596_{\pm 10}$	$1_{\pm 0}$	$5_{\pm 0}$	$27_{\pm 5}$	$1_{\pm 0}$	$14_{\pm 0}$	$153_{\pm 3}$	$1_{\pm 0}$	$27_{\pm 1}$	$583_{\pm 9}$	$1_{\pm 0}$	$16_{\pm 1}$	$263_{\pm 6}$
PCA	$1_{\pm 0}$	$0_{\pm 0}$	—	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

13 Data Set II

This data sets was generate as described in Section 12, but instead of drawing the remaining components of the spanning outer product vectors from $\mathcal{N}(0, 0.01)$, they were now drawn from $\mathcal{N}(0, 0.5)$.

Table 6: Comparison for 50 factors / hidden units extracted by RFN, RFN without normalization (RFNn), denoising autoencoder (DAE), restricted Boltzmann machines (RBM), factor analysis with a very sparse prior (FAsp), factor analysis with a Laplace prior (FAlap), independent component analysis (ICA), sparse factor analysis (SFA), factor analysis (FA), and principal component analysis (PCA) on nine data sets. Criteria are: sparseness of the coding units (SP), reconstruction error (ER), and the difference between the empirical and the model covariance matrix (CO). The lower right column block gives the average SP (%), ER and CO. Results reported here, are the mean together with the standard deviation of 100 instances. The maximal value in the table and the maximal standard deviation was set to 999 and to 99, respectively.

	D1			D2			D3			D4			D5		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$72_{\pm 1}$	$74_{\pm 2}$	$11_{\pm 1}$	$75_{\pm 0}$	$240_{\pm 3}$	$72_{\pm 2}$	$75_{\pm 0}$	$462_{\pm 5}$	$260_{\pm 6}$	$72_{\pm 1}$	$79_{\pm 2}$	$12_{\pm 1}$	$75_{\pm 0}$	$244_{\pm 3}$	$75_{\pm 2}$
RFNn	$68_{\pm 1}$	$122_{\pm 5}$	$32_{\pm 4}$	$74_{\pm 0}$	$285_{\pm 4}$	$97_{\pm 3}$	$74_{\pm 0}$	$537_{\pm 7}$	$331_{\pm 8}$	$65_{\pm 1}$	$144_{\pm 6}$	$48_{\pm 6}$	$74_{\pm 0}$	$290_{\pm 4}$	$102_{\pm 4}$
DAE	$61_{\pm 0}$	$82_{\pm 2}$	—	$66_{\pm 0}$	$243_{\pm 2}$	—	$66_{\pm 0}$	$461_{\pm 4}$	—	$60_{\pm 0}$	$88_{\pm 2}$	—	$66_{\pm 0}$	$247_{\pm 3}$	—
RBM	$22_{\pm 1}$	$106_{\pm 3}$	—	$11_{\pm 1}$	$301_{\pm 3}$	—	$10_{\pm 1}$	$566_{\pm 6}$	—	$22_{\pm 1}$	$113_{\pm 3}$	—	$11_{\pm 1}$	$308_{\pm 4}$	—
FAsp	$37_{\pm 1}$	$469_{\pm 38}$	$999_{\pm 99}$	$40_{\pm 1}$	$999_{\pm 50}$	$999_{\pm 99}$	$40_{\pm 2}$	$999_{\pm 99}$	$999_{\pm 99}$	$37_{\pm 1}$	$610_{\pm 44}$	$999_{\pm 99}$	$40_{\pm 1}$	$999_{\pm 58}$	$999_{\pm 99}$
FAlap	$4_{\pm 0}$	$50_{\pm 1}$	$392_{\pm 66}$	$4_{\pm 0}$	$228_{\pm 5}$	$135_{\pm 13}$	$5_{\pm 0}$	$443_{\pm 9}$	$406_{\pm 18}$	$4_{\pm 0}$	$51_{\pm 1}$	$477_{\pm 63}$	$4_{\pm 0}$	$230_{\pm 6}$	$147_{\pm 18}$
ICA	$2_{\pm 0}$	$35_{\pm 0}$	—	$2_{\pm 0}$	$168_{\pm 2}$	—	$2_{\pm 0}$	$327_{\pm 4}$	—	$2_{\pm 0}$	$35_{\pm 0}$	—	$2_{\pm 0}$	$170_{\pm 2}$	—
SFA	$1_{\pm 0}$	$42_{\pm 1}$	$26_{\pm 3}$	$1_{\pm 0}$	$210_{\pm 5}$	$61_{\pm 2}$	$1_{\pm 0}$	$409_{\pm 8}$	$220_{\pm 6}$	$1_{\pm 0}$	$41_{\pm 1}$	$32_{\pm 4}$	$1_{\pm 0}$	$211_{\pm 5}$	$63_{\pm 2}$
FA	$1_{\pm 0}$	$42_{\pm 1}$	$13_{\pm 2}$	$1_{\pm 0}$	$210_{\pm 4}$	$58_{\pm 2}$	$1_{\pm 0}$	$409_{\pm 8}$	$214_{\pm 6}$	$1_{\pm 0}$	$41_{\pm 1}$	$17_{\pm 2}$	$1_{\pm 0}$	$212_{\pm 5}$	$60_{\pm 2}$
PCA	$0_{\pm 0}$	$35_{\pm 0}$	—	$0_{\pm 0}$	$168_{\pm 2}$	—	$0_{\pm 0}$	$327_{\pm 4}$	—	$0_{\pm 0}$	$35_{\pm 0}$	—	$0_{\pm 0}$	$170_{\pm 2}$	—

	D6			D7			D8			D9			average		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$75_{\pm 0}$	$464_{\pm 5}$	$264_{\pm 6}$	$73_{\pm 0}$	$68_{\pm 2}$	$9_{\pm 1}$	$75_{\pm 0}$	$237_{\pm 3}$	$69_{\pm 1}$	$75_{\pm 0}$	$459_{\pm 5}$	$257_{\pm 6}$	$74_{\pm 0}$	$259_{\pm 3}$	$114_{\pm 3}$
RFNn	$74_{\pm 0}$	$541_{\pm 6}$	$336_{\pm 8}$	$71_{\pm 1}$	$106_{\pm 4}$	$23_{\pm 3}$	$74_{\pm 0}$	$279_{\pm 3}$	$91_{\pm 2}$	$75_{\pm 0}$	$533_{\pm 6}$	$325_{\pm 8}$	$72_{\pm 1}$	$315_{\pm 5}$	$154_{\pm 5}$
DAE	$66_{\pm 0}$	$465_{\pm 4}$	—	$62_{\pm 0}$	$75_{\pm 2}$	—	$66_{\pm 0}$	$238_{\pm 2}$	—	$66_{\pm 0}$	$458_{\pm 4}$	—	$64_{\pm 0}$	$262_{\pm 3}$	—
RBM	$10_{\pm 1}$	$570_{\pm 6}$	—	$20_{\pm 1}$	$97_{\pm 3}$	—	$11_{\pm 1}$	$294_{\pm 3}$	—	$10_{\pm 1}$	$562_{\pm 5}$	—	$14_{\pm 1}$	$324_{\pm 4}$	—
FAsp	$41_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$38_{\pm 1}$	$335_{\pm 32}$	$999_{\pm 99}$	$41_{\pm 1}$	$999_{\pm 40}$	$999_{\pm 99}$	$41_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$39_{\pm 1}$	$999_{\pm 69}$	$999_{\pm 99}$
FAlap	$5_{\pm 0}$	$447_{\pm 9}$	$413_{\pm 19}$	$4_{\pm 0}$	$49_{\pm 1}$	$292_{\pm 57}$	$4_{\pm 0}$	$227_{\pm 5}$	$123_{\pm 11}$	$5_{\pm 0}$	$443_{\pm 9}$	$401_{\pm 17}$	$4_{\pm 0}$	$241_{\pm 5}$	$310_{\pm 31}$
ICA	$2_{\pm 0}$	$329_{\pm 4}$	—	$2_{\pm 0}$	$35_{\pm 0}$	—	$2_{\pm 0}$	$167_{\pm 2}$	—	$2_{\pm 0}$	$325_{\pm 4}$	—	$2_{\pm 0}$	$177_{\pm 2}$	—
SFA	$1_{\pm 0}$	$412_{\pm 8}$	$223_{\pm 7}$	$1_{\pm 0}$	$42_{\pm 1}$	$19_{\pm 3}$	$1_{\pm 0}$	$209_{\pm 4}$	$59_{\pm 2}$	$1_{\pm 0}$	$408_{\pm 9}$	$218_{\pm 7}$	$1_{\pm 0}$	$221_{\pm 5}$	$102_{\pm 4}$
FA	$1_{\pm 0}$	$412_{\pm 8}$	$217_{\pm 7}$	$1_{\pm 0}$	$42_{\pm 1}$	$10_{\pm 1}$	$1_{\pm 0}$	$209_{\pm 4}$	$57_{\pm 2}$	$1_{\pm 0}$	$409_{\pm 9}$	$213_{\pm 7}$	$1_{\pm 0}$	$221_{\pm 5}$	$95_{\pm 3}$
PCA	$0_{\pm 0}$	$329_{\pm 4}$	—	$0_{\pm 0}$	$35_{\pm 0}$	—	$0_{\pm 0}$	$167_{\pm 2}$	—	$0_{\pm 0}$	$325_{\pm 4}$	—	$0_{\pm 0}$	$177_{\pm 2}$	—

Table 7: Comparison for 100 factors / hidden units extracted by RFN, RFN without normalization (RFNn), denoising autoencoder (DAE), restricted Boltzmann machines (RBM), factor analysis with a very sparse prior (FAsp), factor analysis with a Laplace prior (FAlap), independent component analysis (ICA), sparse factor analysis (SFA), factor analysis (FA), and principal component analysis (PCA) on nine data sets. Criteria are: sparseness of the coding units (SP), reconstruction error (ER), and the difference between the empirical and the model covariance matrix (CO). The lower right column block gives the average SP (%), ER and CO. Results reported here, are the mean together with the standard deviation of 100 instances. The maximal value in the table and the maximal standard deviation was set to 999 and to 99, respectively.

	D1			D2			D3			D4			D5		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$76_{\pm 1}$	$34_{\pm 3}$	$4_{\pm 1}$	$82_{\pm 1}$	$67_{\pm 8}$	$18_{\pm 3}$	$82_{\pm 1}$	$124_{\pm 16}$	$63_{\pm 12}$	$75_{\pm 1}$	$38_{\pm 3}$	$5_{\pm 1}$	$82_{\pm 1}$	$69_{\pm 10}$	$19_{\pm 5}$
RFNn	$71_{\pm 1}$	$110_{\pm 7}$	$25_{\pm 4}$	$79_{\pm 0}$	$180_{\pm 5}$	$42_{\pm 2}$	$80_{\pm 0}$	$331_{\pm 8}$	$139_{\pm 7}$	$65_{\pm 2}$	$143_{\pm 9}$	$47_{\pm 8}$	$79_{\pm 0}$	$185_{\pm 5}$	$45_{\pm 3}$
DAE	$63_{\pm 0}$	$66_{\pm 2}$	—	$70_{\pm 0}$	$142_{\pm 2}$	—	$70_{\pm 0}$	$264_{\pm 3}$	—	$62_{\pm 0}$	$73_{\pm 2}$	—	$70_{\pm 0}$	$146_{\pm 2}$	—
RBM	$12_{\pm 1}$	$100_{\pm 3}$	—	$5_{\pm 0}$	$282_{\pm 4}$	—	$4_{\pm 0}$	$522_{\pm 6}$	—	$12_{\pm 1}$	$106_{\pm 3}$	—	$5_{\pm 1}$	$288_{\pm 4}$	—
FAsp	$71_{\pm 0}$	$474_{\pm 38}$	$999_{\pm 99}$	$62_{\pm 0}$	$999_{\pm 53}$	$999_{\pm 99}$	$56_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$70_{\pm 0}$	$616_{\pm 44}$	$999_{\pm 99}$	$62_{\pm 0}$	$999_{\pm 60}$	$999_{\pm 99}$
FAlap	$6_{\pm 0}$	$21_{\pm 2}$	$425_{\pm 28}$	$6_{\pm 0}$	$40_{\pm 2}$	$827_{\pm 35}$	$6_{\pm 0}$	$75_{\pm 6}$	$999_{\pm 99}$	$6_{\pm 0}$	$23_{\pm 2}$	$523_{\pm 32}$	$6_{\pm 0}$	$42_{\pm 3}$	$865_{\pm 43}$
ICA	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$10_{\pm 0}$	$71_{\pm 7}$	$1_{\pm 0}$	$15_{\pm 0}$	$84_{\pm 4}$	$1_{\pm 0}$	$28_{\pm 1}$	$254_{\pm 8}$	$1_{\pm 0}$	$12_{\pm 0}$	$87_{\pm 8}$	$1_{\pm 0}$	$16_{\pm 0}$	$92_{\pm 5}$
FA	$1_{\pm 0}$	$10_{\pm 0}$	$48_{\pm 5}$	$1_{\pm 0}$	$15_{\pm 0}$	$59_{\pm 3}$	$1_{\pm 0}$	$28_{\pm 1}$	$189_{\pm 7}$	$1_{\pm 0}$	$12_{\pm 1}$	$61_{\pm 6}$	$1_{\pm 0}$	$16_{\pm 0}$	$64_{\pm 3}$
PCA	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$3_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

	D6			D7			D8			D9			average		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$82_{\pm 1}$	$127_{\pm 17}$	$65_{\pm 14}$	$77_{\pm 1}$	$30_{\pm 3}$	$3_{\pm 1}$	$82_{\pm 1}$	$64_{\pm 8}$	$17_{\pm 4}$	$82_{\pm 1}$	$123_{\pm 15}$	$62_{\pm 13}$	$80_{\pm 1}$	$75_{\pm 9}$	$28_{\pm 6}$
RFNn	$80_{\pm 0}$	$334_{\pm 8}$	$141_{\pm 7}$	$74_{\pm 1}$	$86_{\pm 4}$	$14_{\pm 2}$	$79_{\pm 0}$	$174_{\pm 4}$	$39_{\pm 2}$	$80_{\pm 0}$	$329_{\pm 7}$	$137_{\pm 6}$	$76_{\pm 1}$	$208_{\pm 6}$	$70_{\pm 5}$
DAE	$70_{\pm 0}$	$266_{\pm 2}$	—	$64_{\pm 0}$	$57_{\pm 2}$	—	$70_{\pm 0}$	$138_{\pm 1}$	—	$70_{\pm 0}$	$262_{\pm 2}$	—	$68_{\pm 0}$	$157_{\pm 2}$	—
RBM	$4_{\pm 0}$	$527_{\pm 6}$	—	$11_{\pm 1}$	$92_{\pm 2}$	—	$4_{\pm 0}$	$274_{\pm 4}$	—	$4_{\pm 0}$	$518_{\pm 6}$	—	$7_{\pm 1}$	$301_{\pm 4}$	—
FAsp	$56_{\pm 0}$	$999_{\pm 99}$	$999_{\pm 99}$	$71_{\pm 0}$	$338_{\pm 33}$	$999_{\pm 99}$	$62_{\pm 1}$	$999_{\pm 42}$	$999_{\pm 99}$	$56_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$63_{\pm 0}$	$999_{\pm 74}$	$999_{\pm 99}$
FAlap	$6_{\pm 0}$	$75_{\pm 6}$	$999_{\pm 89}$	$6_{\pm 0}$	$18_{\pm 2}$	$337_{\pm 24}$	$6_{\pm 0}$	$40_{\pm 3}$	$793_{\pm 37}$	$6_{\pm 0}$	$74_{\pm 6}$	$999_{\pm 89}$	$6_{\pm 0}$	$45_{\pm 3}$	$999_{\pm 53}$
ICA	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$28_{\pm 1}$	$260_{\pm 9}$	$1_{\pm 0}$	$8_{\pm 0}$	$52_{\pm 7}$	$1_{\pm 0}$	$15_{\pm 0}$	$76_{\pm 3}$	$1_{\pm 0}$	$28_{\pm 1}$	$248_{\pm 7}$	$1_{\pm 0}$	$18_{\pm 1}$	$136_{\pm 6}$
FA	$1_{\pm 0}$	$28_{\pm 1}$	$193_{\pm 8}$	$1_{\pm 0}$	$8_{\pm 0}$	$33_{\pm 5}$	$1_{\pm 0}$	$15_{\pm 0}$	$54_{\pm 2}$	$1_{\pm 0}$	$28_{\pm 1}$	$185_{\pm 6}$	$1_{\pm 0}$	$18_{\pm 1}$	$99_{\pm 5}$
PCA	$1_{\pm 0}$	$0_{\pm 0}$	—	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

Table 8: Comparison for 150 factors / hidden units extracted by RFN, RFN without normalization (RFNn), denoising autoencoder (DAE), restricted Boltzmann machines (RBM), factor analysis with a very sparse prior (FAsp), factor analysis with a Laplace prior (FAlap), independent component analysis (ICA), sparse factor analysis (SFA), factor analysis (FA), and principal component analysis (PCA) on nine data sets. Criteria are: sparseness of the factors (SP) reported in %, reconstruction error (ER), and the difference between the empirical and the model covariance matrix (CO). The lower right column block gives the average SP (%), ER and CO. Results reported here, are the mean together with the standard deviation of 100 instances. The maximal value in the table and the maximal standard deviation was set to 999 and to 99, respectively.

	D1			D2			D3			D4			D5		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$81_{\pm 1}$	$12_{\pm 2}$	$1_{\pm 1}$	$86_{\pm 0}$	$16_{\pm 1}$	$4_{\pm 1}$	$86_{\pm 0}$	$29_{\pm 4}$	$15_{\pm 5}$	$80_{\pm 1}$	$15_{\pm 5}$	$2_{\pm 2}$	$86_{\pm 1}$	$17_{\pm 5}$	$5_{\pm 3}$
RFNn	$72_{\pm 1}$	$100_{\pm 8}$	$19_{\pm 4}$	$80_{\pm 0}$	$137_{\pm 4}$	$24_{\pm 1}$	$81_{\pm 0}$	$254_{\pm 6}$	$83_{\pm 4}$	$66_{\pm 0}$	$113_{\pm 3}$	$52_{\pm 5}$	$80_{\pm 0}$	$141_{\pm 4}$	$26_{\pm 2}$
DAE	$64_{\pm 0}$	$62_{\pm 2}$	—	$71_{\pm 0}$	$125_{\pm 2}$	—	$72_{\pm 0}$	$232_{\pm 2}$	—	$63_{\pm 0}$	$69_{\pm 2}$	—	$71_{\pm 0}$	$129_{\pm 2}$	—
RBM	$8_{\pm 0}$	$101_{\pm 3}$	—	$4_{\pm 0}$	$282_{\pm 4}$	—	$3_{\pm 0}$	$521_{\pm 6}$	—	$8_{\pm 0}$	$106_{\pm 3}$	—	$4_{\pm 0}$	$289_{\pm 4}$	—
FAsp	$81_{\pm 1}$	$474_{\pm 38}$	$999_{\pm 99}$	$79_{\pm 0}$	$999_{\pm 53}$	$999_{\pm 99}$	$77_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$80_{\pm 1}$	$616_{\pm 44}$	$999_{\pm 99}$	$79_{\pm 1}$	$999_{\pm 60}$	$999_{\pm 99}$
FAlap	$4_{\pm 0}$	$21_{\pm 2}$	$607_{\pm 34}$	$4_{\pm 0}$	$40_{\pm 2}$	$879_{\pm 40}$	$3_{\pm 0}$	$75_{\pm 6}$	$999_{\pm 96}$	$4_{\pm 0}$	$23_{\pm 2}$	$749_{\pm 42}$	$4_{\pm 0}$	$42_{\pm 3}$	$926_{\pm 45}$
ICA	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 2}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$10_{\pm 0}$	$103_{\pm 9}$	$1_{\pm 0}$	$15_{\pm 0}$	$204_{\pm 7}$	$1_{\pm 0}$	$28_{\pm 1}$	$656_{\pm 12}$	$1_{\pm 0}$	$12_{\pm 0}$	$126_{\pm 10}$	$1_{\pm 0}$	$16_{\pm 0}$	$220_{\pm 8}$
FA	$1_{\pm 0}$	$10_{\pm 0}$	$87_{\pm 8}$	$1_{\pm 0}$	$15_{\pm 0}$	$187_{\pm 5}$	$1_{\pm 0}$	$28_{\pm 1}$	$611_{\pm 11}$	$1_{\pm 0}$	$12_{\pm 1}$	$108_{\pm 9}$	$1_{\pm 0}$	$16_{\pm 0}$	$200_{\pm 6}$
PCA	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$3_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

	D6			D7			D8			D9			average		
	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO	SP	ER	CO
RFN	$86_{\pm 1}$	$29_{\pm 7}$	$15_{\pm 6}$	$82_{\pm 1}$	$10_{\pm 3}$	$1_{\pm 1}$	$86_{\pm 1}$	$17_{\pm 10}$	$5_{\pm 9}$	$86_{\pm 1}$	$31_{\pm 19}$	$16_{\pm 13}$	$84_{\pm 1}$	$20_{\pm 6}$	$7_{\pm 4}$
RFNn	$81_{\pm 0}$	$255_{\pm 6}$	$84_{\pm 3}$	$76_{\pm 1}$	$74_{\pm 5}$	$9_{\pm 2}$	$81_{\pm 0}$	$133_{\pm 3}$	$23_{\pm 1}$	$81_{\pm 0}$	$250_{\pm 7}$	$81_{\pm 4}$	$77_{\pm 0}$	$162_{\pm 5}$	$45_{\pm 3}$
DAE	$72_{\pm 0}$	$234_{\pm 2}$	—	$65_{\pm 0}$	$53_{\pm 2}$	—	$72_{\pm 0}$	$122_{\pm 1}$	—	$72_{\pm 0}$	$230_{\pm 2}$	—	$69_{\pm 0}$	$140_{\pm 2}$	—
RBM	$3_{\pm 0}$	$525_{\pm 6}$	—	$8_{\pm 0}$	$93_{\pm 3}$	—	$3_{\pm 0}$	$273_{\pm 4}$	—	$3_{\pm 0}$	$517_{\pm 6}$	—	$5_{\pm 0}$	$301_{\pm 4}$	—
FAsp	$77_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$81_{\pm 1}$	$338_{\pm 33}$	$673_{\pm 99}$	$79_{\pm 0}$	$999_{\pm 42}$	$999_{\pm 99}$	$77_{\pm 1}$	$999_{\pm 99}$	$999_{\pm 99}$	$79_{\pm 1}$	$999_{\pm 74}$	$999_{\pm 99}$
FAlap	$3_{\pm 0}$	$75_{\pm 6}$	$999_{\pm 94}$	$4_{\pm 0}$	$18_{\pm 2}$	$479_{\pm 31}$	$4_{\pm 0}$	$40_{\pm 3}$	$831_{\pm 43}$	$3_{\pm 0}$	$74_{\pm 6}$	$999_{\pm 95}$	$4_{\pm 0}$	$45_{\pm 3}$	$999_{\pm 58}$
ICA	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—	$3_{\pm 1}$	$0_{\pm 0}$	—
SFA	$1_{\pm 0}$	$28_{\pm 1}$	$668_{\pm 12}$	$1_{\pm 0}$	$8_{\pm 0}$	$78_{\pm 8}$	$1_{\pm 0}$	$15_{\pm 0}$	$188_{\pm 5}$	$1_{\pm 0}$	$28_{\pm 1}$	$644_{\pm 9}$	$1_{\pm 0}$	$18_{\pm 1}$	$321_{\pm 9}$
FA	$1_{\pm 0}$	$28_{\pm 1}$	$622_{\pm 11}$	$1_{\pm 0}$	$8_{\pm 0}$	$64_{\pm 7}$	$1_{\pm 0}$	$15_{\pm 0}$	$173_{\pm 4}$	$1_{\pm 0}$	$28_{\pm 1}$	$599_{\pm 9}$	$1_{\pm 0}$	$18_{\pm 1}$	$294_{\pm 8}$
PCA	$1_{\pm 0}$	$0_{\pm 0}$	—	$4_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—	$1_{\pm 0}$	$0_{\pm 0}$	—	$2_{\pm 0}$	$0_{\pm 0}$	—

14 RFN Pretraining for Convolution Nets

We assess the performance of RFN *first layer* pretraining on *CIFAR-10* and *CIFAR-100* for three deep convolutional network architectures: (i) the AlexNet Krizhevsky et al. [2012], (ii) Deeply Supervised Networks (DSN) Lee et al. [2014], and (iii) our 5-Convolution-Network-In-Network (5C-NIN).

Both CIFAR datasets contain 60k 32x32 RGB-color images, which were divided into 50k train and 10k test sets, split between 10 (CIFAR10) and 100 (CIFAR100) categories. Both datasets are preprocessed as described in Goodfellow et al. [2013] by global contrast normalization and ZCA whitening. Additionally, the datasets were augmented by padding the images with four zero pixels at all borders. For data augmentation, at the beginning of every epoch, images in the training set were distorted by random translation and random flipping in horizontal and vertical directions. For the AlexNet, we neither preprocessed nor augmented the datasets.

Inspired by Lin et al. [2013]'s Network In Network, we constructed a 5-Convolution-Network-In-Network (5C-NIN) architecture with five convolutional layers, each followed by a 2x2 max-pooling layer (stride 1) and a multilayer perceptron (MLP) convolutional layer. ReLUs were used for the convolutional layers and dropout for regularization. We followed Krizhevsky [2009] for weight initialization, learning rates, and learning policies. The networks were trained using mini-batches of size 100 and 128 for 5C-NIN and AlexNet, respectively.

For RFN pretraining, we randomly extracted 5x5 patches from the training data to construct 192 filters for DSN and 5C-NIN while 32 for AlexNet. These filters constitute the first convolutional layer of each network which is then trained using default setting. For assessing the improvement by RFNs, we repeated training with randomly initialized weights in the first layer. The results are presented in Tab. 9. For comparison, the lower panel of the table reports the performance of the currently top performing networks: Network In Network (NIN, Lin et al. [2013]), Maxout Networks (MN, Goodfellow et al. [2013]) and DeepCNiN Graham [2014]. *In all cases pretraining with RFNs decreases the test error rate.*

Table 9: The upper panel shows results of convolutional deep networks with first layer pretrained by RFN ("RFN") and with first layer randomly initialized ("org"). The first column gives the network architecture, namely, AlexNet, Deeply Supervised Networks (DSN), and our 5-Convolution-Network-In-Network (5C-NIN). The test error rates are reported (for CIFAR-100 DSN model was missing). Currently best performing networks Network In Network (NIN), Maxout Networks (MN), and DeepCNiN are reported in the lower panel. In all cases pretraining with RFNs decreased the test error rate.

Dataset	CIFAR-10		CIFAR-100		
	org	RFN	org	RFN	augmented
AlexNet	18.21	18.04	46.18	45.80	
DSN	7.97	7.74	34.57	-	$\surd$
5C-NIN	7.81	7.63	29.96	29.75	$\surd$
NIN	8.81	-	35.68	-	$\surd$
MN	9.38	-	38.57	-	$\surd$
DeepCNiN	6.28	-	24.30	-	$\surd$

15 Running Times for RFN's Projected Newton Step

In this section, we report the running times for RFN's projected Newton step and for solving a quadratic program using NumPy Python and CVXOPT (Python Software for Convex Optimization), respectively. Both benchmarks were profiled with the same hardware using only the CPU. Fig. 1 shows the run times for various problem sizes in [s] both approaches. The projected Newton step complexity per iteration is $O(nl)$, see Fig. 2. In contrast, a quadratic program solver typically requires for the (nl) variables (the means of the hidden units for all samples) $O(n^4 l^4)$ steps to find the minimum Ben-Tal and Nemirovski [2001].

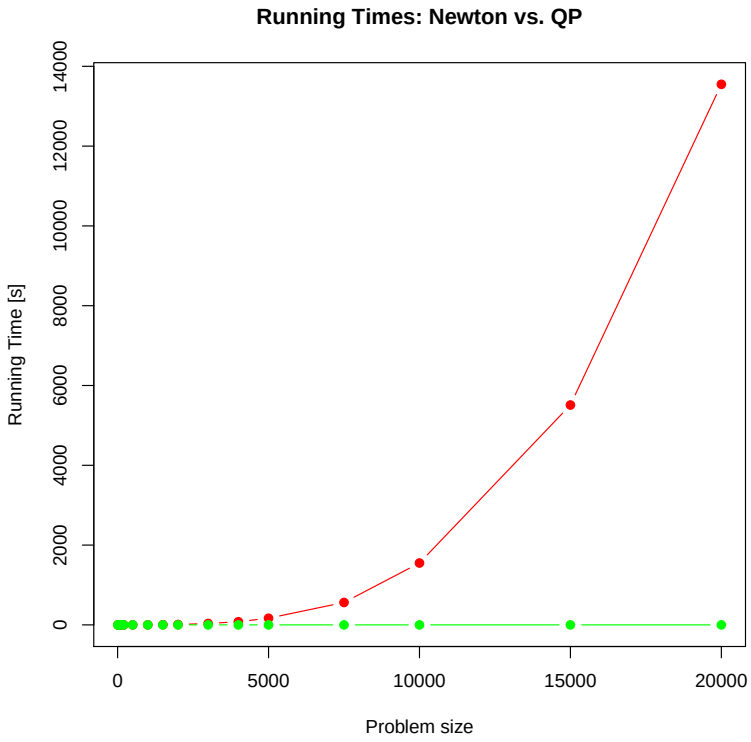

Figure 1: Running times for various problem sizes in [s] of RFN's projected Newton step and of quadratic program solver.

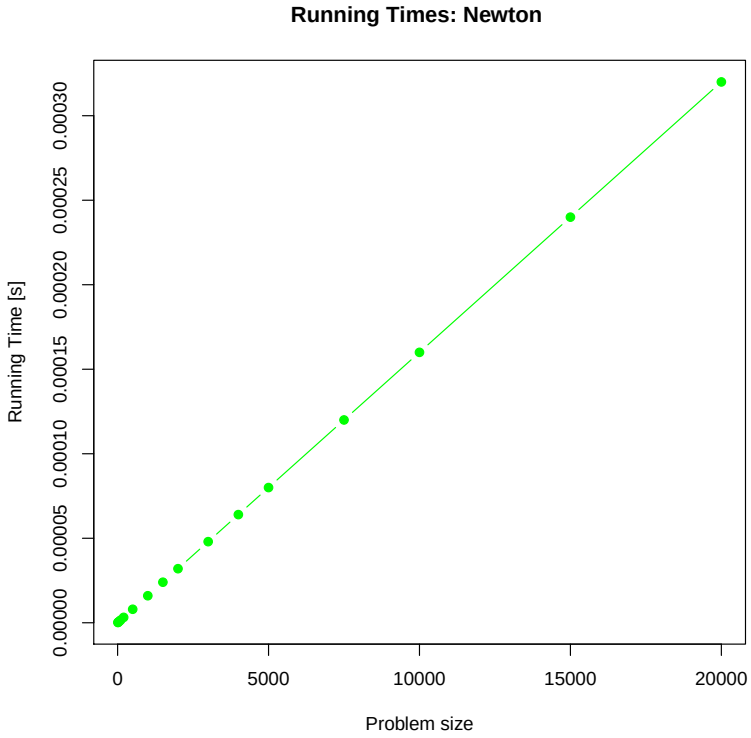

Figure 2: Running times for various problem sizes in [s] of RFN's projected Newton step.

References

J. Abadie and J. Carpentier. *Optimization*, chapter Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints. Academic Press, 1969.

A. Ben-Tal and A. Nemirovski. *Interior Point Polynomial Time Methods for Linear Programming, Conic Quadratic Programming, and Semidefinite Programming*, chapter 6, pages 377–442. Society for Industrial and Applied Mathematics, 2001. doi: 10.1137/1.9780898718829.ch6.

D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Automat. Control*, 21:174–184, 1976.

D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM J. Control Optim.*, 20:221–246, 1982.

E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *Siam Journal on Optimization*, 10(4):1196–1211, 2000. doi: 10.1137/S1052623497330963.

D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2012*, 2012. Long preprint arXiv:1202.2745v1 [cs.CV].

M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning (ICML08)*, volume 25, pages 264–271. ACM New York, 2008. doi: 10.1145/1390156.1390190.

M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification for text categorization. *Journal of Machine Learning Research*, 13(1):1891–1926, 2012.

R. P. Feynman. *Statistical Mechanics*. Benjamin, Reading, MA, 1972.

B. J. Frey and G. E. Hinton. Variational learning in nonlinear Gaussian belief networks. *Neural Computation*, 11(1):193–214, 1999.

K. Friston. A free energy principle for biological systems. *Entropy*, 14:2100–2121, 2012.

K. Ganchev, J. Graca, J. Gillenwater, and B. Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049, 2010.

I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *ArXiv e-prints*, 2013.

J. V. Graca, , K. Ganchev, and B. Taskar. Expectation maximization and posterior constraints. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 569–576, 2007.

J. V. Graca, , K. Ganchev, B. Taskar, and F. Pereira. Posterior vs. parameter sparsity in latent variable models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 664–672, 2009.

Benjamin Graham. Fractional max-pooling. *CoRR*, abs/1412.6071, 2014. URL `http://arxiv.org/abs/1412.6071`.

A. Gunawardana and W. Byrne. Convergence theorems for generalized alternating minimization procedures. *Journal of Machine Learning Research*, 6:2049–2073, 2005.

M. Harva and A. Kaban. A variational bayesian method for rectified factor analysis. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN'05)*, pages 185–190, 2005.

M. Harva and A. Kaban. Variational learning for rectified factor analysis. *Signal Processing*, 87(3): 509–527, 2007. doi: 10.1016/j.sigpro.2006.06.006.

E. J. Haug and J. S. Arora. *Applied optimal design*. J. Wiley & Sons, New York, 1979.

A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Comput.*, 9(7):1483–1492, 1999.

C. T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1999.

D. Kim, S. Sra, and I. S. Dhillon. A new projected quasi-Newton approach for the nonnegative least squares problem. Technical Report TR-06-54, Department of Computer Sciences, University of Texas at Austin, 2006.

A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Deptartment of Computer Science, University of Toronto, 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-Supervised Nets. *ArXiv e-prints*, 2014.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013. URL `http://arxiv.org/abs/1312.4400`.

R. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, Cambridge, MA, 1998.

R. Patel and M. Toda. Trace inequalities involving hermitian matrices. *Linear Algebra and its Applications*, 23:13–20, 1979. doi: 10.1016/0024-3795(79)90089-2.

J. B. Rosen. The gradient projection method for nonlinear programming. part ii. nonlinear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, 1961. doi: 10.1137/0109044.

T. Serafini, G. Zanghirati, and L. Zanni. Gradient projection methods for quadratic programs and applications in training support vector machines. *Optimization Methods and Software*, 20(2-3): 353–378, 2005. doi: 10.1080/10556780512331318182.

N. Srebro. *Learning with Matrix Factorizations*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2004.

C. F. J. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95–103, 1983.

W. I. Zangwill. *Nonlinear Programming: A Unified Approach*. Prentice Hall, Englewood Cliffs, N.J., 1969.