# Supplementary Material

## A. Scaling UP LDS Learning to Text

As discussed in Section 4.3, we whiten our data using

$$W = \Psi_0^{-\frac{1}{2}} = \text{diag}(\mu_1^{-\frac{1}{2}}, \ldots, \mu_V^{-\frac{1}{2}}). \tag{20}$$

Besides improving the empirical performance of SSID, working in the whitened coordinate system also simplifies various details used in Section 4 when scaling up LDS learning for text. Under this transformation, we have $\Psi_0 = \text{diag}(\mu) - \mu\mu^\top$. This simplifies various steps because our estimators (12) and (9) are of the form $I$ − [low rank matrix], rather than diag($\mu$) − [low rank matrix]. In the whitened coordinates, the data are orthogonal to $\mu^{\frac{1}{2}}$, rather than **1**.

### A.1. Recovering PSD $D$ in SSID

While SSID is consistent, for finite data the procedure is not guaranteed to yield a positive semidefinite (PSD) estimate for $D$, which is required because it is a covariance matrix. In our particular case, the $D$ we seek will be singular on the span of $\mu^{\frac{1}{2}}$, but Subspace ID will still not guarantee that $D$ will be PSD on ${\mu^{\frac{1}{2}}}^\perp$.

This is critical because if $D$ is not PSD on this subspace, then we can not define a valid Kalman filtering procedure for the model (see Sec. A.2). However, due to the structure of our data distribution, $D$ can easily be fixed post-hoc.

From (12) we have the estimator

$$D = I - \mu^{\frac{1}{2}}{\mu^{\frac{1}{2}}}^\top - C\Sigma_1 C^\top \tag{21}$$

Next, define $D_\alpha = I - \mu^{\frac{1}{2}}{\mu^{\frac{1}{2}}}^\top - (1-\alpha)C\Sigma_1 C^\top$ and define the PSD estimator $D' = D_{\alpha_0}$, where $\alpha_0$ is the minimal value such that $D_\alpha$ is PSD on ${\mu^{\frac{1}{2}}}^\perp$. We next show how to find $\alpha_0$.

We have that $D_\alpha$ is PSD on ${\boldsymbol{\mu}^{\frac{1}{2}}}^\perp$ iff the maximum eigenvalue of $(1-\alpha)C\Sigma_1 C^\top$ is less than 1. This is because $\mu^{\frac{1}{2}}$ is a unit vector and we can ignore any cross terms between $\mu^{\frac{1}{2}}{\mu^{\frac{1}{2}}}^\top$ and $(1-\alpha)C\Sigma_1 C^\top$ because col$(C) = {\mu^{\frac{1}{2}}}^\perp$, which is true because the data lies in this subspace. Therefore we can find $\alpha_0$ using the following procedure:

1. Find $s_0$, the maximal eigenvalue of $C\Sigma_1 C^\top$, using power iteration. This can be done efficiently by keep $C\Sigma_1 C^\top$ in its factorized form and not instantiating a $V \times V$ matrix.
2. If $s_0 < 1$, set $\alpha_0 = 0$. Otherwise, set $\alpha_0 = \frac{s_0 - 1}{s_0}$.

### A.2. Efficiently Computing the Kalman Gain Matrix

Next, recall our expression (6) for the steady state Kalman gain $K = \Sigma_1 C^\top S_{ss}^{-1}$, which comes from solving the system

$$KS_{ss} = \Sigma_1 C^\top, \tag{22}$$

where

$$S_{ss} = C\Sigma_1 C^\top + D \tag{23}$$

Furthermore, note that both of our estimators for $D$, (12) and (9), maintain the property that $\mu^{\frac{1}{2}}$ is an eigenvector of eigenvalue 0 for $D$.

Since $\mu^{\frac{1}{2}}$ is also orthogonal to col$(C)$, we have that $\mu^{\frac{1}{2}} \notin$ Col$(S_{ss})$. Therefore, we cannot use (6) directly because $S_{ss}$ is not invertible along this direction. However, we can still solve (22) as $K = \Sigma_1 C^\top S_{ss}^+$. This pseudoinverse can be characterized as:

$$S_{ss}^+ = [\text{inversion of } S_{ss} \text{ within col}(S_{ss})]\ [\text{projection onto col}(S_{ss})] \tag{24}$$

Furthermore, note that both estimators for D have the form that

$$
\begin{aligned}
D &= \Psi_0 - (\text{PSD, low rank, and } \perp \mu^{\frac{1}{2}}) && (25)\\
&= I - \mu^{\frac{1}{2}}\mu^{\frac{1}{2}^\top} - (\text{PSD, low rank and } \perp \mu^{\frac{1}{2}}) && (26)\\
&:= I - \mu^{\frac{1}{2}}\mu^{\frac{1}{2}^\top} - L && (27)
\end{aligned}
$$

Therefore, it remains to define the pseudoinverse of

$$S_{ss} = I - \mu^{\frac{1}{2}}\mu^{\frac{1}{2}^\top} + C(\Sigma_1 - M)C^\top). \qquad (28)$$

Furthermore, since $\text{col}(L) = \text{col}(C) = \mu^{\frac{1}{2}^\perp}$, we can define $L = CMC^\top$ for some positive definite $M$, so we consider

$$S_{ss} = I - \mu^{\frac{1}{2}}\mu^{\frac{1}{2}^\top} + C(\Sigma_1 - M)C^\top). \qquad (29)$$

Observe that

$$(I + C(\Sigma_1 - M)C^\top)^{-1} \qquad (30)$$

is a valid inverse for $S_{ss}$ on $\mu^{\frac{1}{2}^\perp}$. This follows from the orthogonality of $\mu^{\frac{1}{2}}$ and $\text{col}(C)$, so we can effectively ignore the $\mu^{\frac{1}{2}}$ term in (29) when inverting it on $\mu^{\frac{1}{2}^\perp}$.

Therefore, we employ

$$(S_{ss})^+ = (I + C(\Sigma_1 - M)C^\top)^{-1}(I - \mu^{\frac{1}{2}}\mu^{\frac{1}{2}}), \qquad (31)$$

where the right term is an orthogonal projection onto $\mu^{\frac{1}{2}^\perp}$.

The term in the inverse (31) is diagonal-plus-low-rank and can be manipulated efficiently using the matrix inversion lemma formula (53):

$$(I + C(\Sigma_1 - M)C^\top)^{-1} = I - C((\Sigma_1 - M)^{-1} + C'C)^{-1}C^\top. \qquad (32)$$

Therefore we can obtain $K$ without instantiating an intermediate matrix of size $V \times V$.

Recall the filtering equation (4):

$$\hat{x}_t^t = (A - KCA)\hat{x}_{t-1}^{t-1} + Kw_t.$$

We seek to avoid any $O(V)$ (or worse) computation at test time when filtering. First of all, we can precompute $(A - KCA)$. For the second term, there are only $V$ possible values for the unwhitened input $w_t = \tilde{w}_t - \mu$, so we would like to precompute $KW(\tilde{w}_t - \mu)$ for every possible value that the indicator $\tilde{w}_t$ can take on. Let $\tilde{w}_t = e_i$, we have:

$$
\begin{aligned}
KW(\tilde{w}_t - \mu) &= \Sigma_1 C^\top S_{ss}^+ W(e_i - \mu) && (33)\\
&= \Sigma_1 C^\top (I + C(\Sigma_1 - M)C^\top)^{-1}(I - \mu^{\frac{1}{2}}\mu^{\frac{1}{2}^\top})(We_i - \mu^{\frac{1}{2}}) && (34)\\
&= \Sigma_1 C^\top (I + C(\Sigma_1 - M)C^\top)^{-1}(I - \mu^{\frac{1}{2}}\mu^{\frac{1}{2}^\top})We_i && (35)\\
&= \Sigma_1 C^\top (I + C(\Sigma_1 - M)C^\top)^{-1} We_i && (36)\\
&= \left[\Sigma_1 C^\top (I + C(\Sigma_1 - M)C^\top)^{-1} W\right]_i, && (37)\\
& && (38)
\end{aligned}
$$

In the final line, the subscript $i$ denotes the $i$th column of a matrix.

### A.3. Likelihood Computation

$S_{ss}$ is also used when computing the log-likelihood of input data $(w_1, \dots, w_T)$:

$$LL = -TV \log(2\pi) - \frac{1}{2} \log\det(S_{ss}) + \sum_{t=1}^{\top} (w_t^{pred} - w_t)^\top S_{ss}^{-1} (w_t^{pred} - w_t). \tag{39}$$

Here, $w_t^{pred} = CA\hat{x}_t$, where $\hat{x}_t$ is the posterior mean for $x_t$ given observations $w_{1:(t-1)}$. $S_{ss}$ is only invertible along ${\mu^{\frac{1}{2}}}^{\perp}$, but $(w_t^{pred} - w_t)$ varies only on this subspace, so we can effectively ignore the zero-variance direction $\mu^{\frac{1}{2}}$. Therefore, we just use (30) as $S_{ss}^{-1}$ in (39).

For the data-dependent term in our likelihood, we have:

$$\begin{align}
& -\frac{1}{2} \sum_{t=1}^{\top} (w_t^{pred} - w_t)^\top S_{ss}^{-1} (w_t^{pred} - w_t) \tag{40} \\
&= \frac{-1}{2} tr \left( S_{ss}^{-1} \mathbb{E}_t[(w_t^{pred} - w_t)(w_t^{pred} - w_t)^\top] \right) \tag{41} \\
&= \frac{-1}{2} tr \left( S_{ss}^{-1} \mathbb{E}_t[(w_t - CA\hat{x}_t)(w_t - CA\hat{x}_t)^\top] \right) \tag{42} \\
&= \frac{-1}{2} \left( tr \left( S_{ss}^{-1} \mathbb{E}_t[w_t w_t^\top] \right) - 2tr \left( S_{ss}^{-1} \mathbb{E}_t[w_t \hat{x}_t^\top] A^\top C^\top \right) + tr \left( S_{ss}^{-1} CA \mathbb{E}_t[\hat{x}_t \hat{x}_t^\top] A^\top C^\top \right) \right) \tag{43} \\
&= \frac{-1}{2} \left( tr \left( S_{ss}^{-1} I \right) - 2tr \left( S_{ss}^{-1} \mathbb{E}_t[w_t \hat{x}_t^\top] A^\top C^\top \right) + tr \left( S_{ss}^{-1} CA \mathbb{E}_t[\hat{x}_t \hat{x}_t^\top] A^\top C^\top \right) \right) \tag{44}
\end{align}$$

Note that the $\mathbb{E}_t[\hat{x}_t \hat{x}_t^\top]$ term above is different from $\Sigma_1$, since the former is from the posterior distribution given the input data and $\Sigma_1$ is from the prior.

The first term can be computed using (57). The latter two terms are of the form $tr\left(S_{ss}^{-1} Z W^\top\right)$, where $Z$ and $W$ are both $V \times k$, so we can invoke (58). For the $\log\det(S_{ss})$ term, we consider $S_{ss}$ only on ${\mu^{\frac{1}{2}}}^{\perp}$, so we compute $-\log\det(S_{ss}^{-1})$, where $S_{ss}^{-1}$ comes from (30) and we employ the formula (55).

## B. Background

### B.1. Non-Steady-State Kalman Filtering and Smoothing

We will use $\hat{x}_t^{\tau}$ and $S_t^{\tau}$ for the mean and variance under the posterior for $x_t$ given $w_{1:\tau}$. We will use $\bar{x}_t$ and $S_t^T$ when considering the posterior for $x_t$ given all the data $w_{1:T}$. The following are the forward 'filtering' steps (Kalman, 1960; Ghahramani & Hinton, 1996):

$$\begin{align}
\hat{x}_t^{t-1} &= A\hat{x}_{t-1}^{t-1} \tag{45} \\
S_t^{t-1} &= AS_{t-1}^{t-1}A^\top + Q \tag{46} \\
K_t &= S_t^{t-1} C' (C S_{t-1}^{t-1} C^\top + D)^{-1} \tag{47} \\
\hat{x}_t^t &= \hat{x}_{t-1}^t + K_t (w_t - C\hat{x}_t^{t-1}) \tag{48} \\
S_t^{t-1} &= S_t^{t-1} - K_t C S_t^{t-1} \tag{49}
\end{align}$$

Next, we have the backwards 'smoothing' steps:

$$\begin{align}
J_{t-1} &= S_{t-1}^{t-1} A' (S_t^{t-1})^{-1} \tag{50} \\
\bar{x}_{t-1} &= \hat{x}_{t-1}^{t-1} + J_{t-1} (\bar{x}_t^T - A\hat{x}_{t-1}^{t-1}) \tag{51} \\
S_{t-1}^T &= S_{t-1}^{t-1} + J_{t-1} (S_t^\top - S_t^{t-1}) J_{t-1}^T \tag{52}
\end{align}$$

Note that the updates for the variances $S$ are data-independent and just depend on the parameters of the model. They will converge quickly to time-independent 'steady state' quantities.

### B.2. Matrix Inversion Lemma

Following Press et al. (1987), we have

$$(A+USV^\top)^{-1} = A^{-1} - A^{-1}U(S^{-1}+V^\top A^{-1}U)^{-1}V^\top A^{-1} \tag{53}$$

and the related expression for determinants:

$$\det(A+USV^\top) = \det(S)\det(A)\det(S^{-1}+V^\top A^{-1}U). \tag{54}$$

i.e.

$$\log\det(A+USV^\top) = \log\det(S) + \log\det(A) + \log\det(S^{-1}+V^\top A^{-1}U). \tag{55}$$

Expression (53) is useful if we already have an inverse for $A$ and want to efficiently compute the inverse of a low-rank perturbation of $A$. It is also useful in order to be able to do linear algebra using $(A+USV^\top)^{-1}$ without actually instantiating a $V \times V$ matrix, which can be unmanageable in terms of both time and space for large $V$. For example, let $M$ be an $V \times m$ matrix with $m << V$, then we can compute $M(A+USV^\top)^{-1}$ using (53) by carefully placing our parentheses such that no $V \times V$ matrix is required. In our application, $A$ is diagonal, so computing its inverse is trivial. Also, note that (53) can be used recursively, if $A$ is defined as another sum of an easily invertible matrix and a low rank matrix.

Along these lines, here are a few additional useful identities that follow from (53) for quantities that can be computed without $V^2$ time or storage. Here, we assume that both $A^{-1}$ and $tr(A^{-1})$ can be computed inexpensively (e.g., $A$ is diagonal).

For any product $XY^\top$, where $X$ and $Y$ are $V \times k$ matrices, note that we can compute $tr(XY^T)$ in $O(Vk)$ time as

$$tr(XY^T) = \sum_i \sum_j X_{ij}Y_{ij}. \tag{56}$$

We can use this to compute the trace of the inverse of a matrix implicitly defined via the matrix inversion lemma:

$$tr\left[(A+USV^\top)^{-1}\right] = tr(A^{-1}) - tr\left[\underbrace{A^{-1}U(S^{-1}+V^\top A^{-1}U)^{-1}}_{X}\underbrace{V^\top A^{-1}}_{Y^\top}\right]. \tag{57}$$

More generally, Let $Z$ and $W$ be $V \times k$ matrices, then we compute

$$tr\left[(A+USV^\top)^{-1}ZW^\top\right] = tr(\underbrace{A^{-1}Z}_{X}\underbrace{W^\top}_{Y^\top}) - tr\left[\underbrace{A^{-1}U(S^{-1}+V^\top A^{-1}U)^{-1}}_{X}\underbrace{V^\top A^{-1}ZW^\top}_{Y^\top}\right] \tag{58}$$

We use (58) when computing the Likelihood in Section A.3.

## C. SSID Initialization vs. Random Initialization

In Figure 2, we contrast the progress of EM, in terms of the log-likelihood of the training data, when initializing with SSID vs. initializing randomly (Random). Note that the initial values of SSID and Random are nearly identical. This is due to model mispecification, and the fact that we chose the lengthscales of the random parameters post-hoc, by looking at the lengthscales of the SSID parameters. Over the course of 100 EM iterations, the model initialized with SSID climbs quickly and begins leveling out, whereas it takes a long time for the Random model to begin climbing at all. We truncate at 100 EM iterations, since we actually use the SSID-initialized model after the 50th iteration. After that, we find that local POS tagging accuracy diminished.

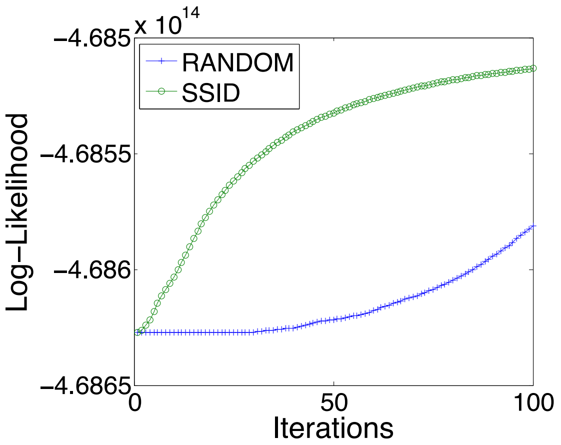

*Figure 2.* EM Log-Likelihood vs. training iterations for random initialization and SSID initialization.