9 Appendix

9.1 Proofs

Lemma 4 $D_F(\hat{\mathbf{z}}||\mathbf{z}) = D_{F^*}(\mathbf{y}||\hat{\mathbf{y}})$ *where* $\hat{\mathbf{y}} = \mathbf{f}(\hat{\mathbf{z}})$ *and* $\mathbf{y} = \mathbf{f}(\mathbf{z})$.

Proof: Recall that $F^*(\mathbf{y}) = \max_{\mathbf{z}} \mathbf{z}^T\mathbf{y} - F(\mathbf{z})$, and solving for the maximum $\mathbf{z}$ we obtain

$$\begin{aligned} \frac{d}{d\mathbf{z}} &= \mathbf{y} - \nabla F(\mathbf{z}) = \mathbf{y} - \mathbf{f}(\mathbf{z}) = 0 \\ &\implies \mathbf{z} = \mathbf{f}^{-1}(\mathbf{y}) \end{aligned}$$

giving

$$F^*(\mathbf{y}) = \mathbf{f}^{-1}(\mathbf{y})^T\mathbf{y} - F(\mathbf{f}^{-1}(\mathbf{y}))$$

Now we can rewrite $D_{F^*}(\mathbf{y}||\hat{\mathbf{y}})$ in terms of F and $\mathbf{f}^{-1} = \mathbf{f}^*$

$$\begin{aligned} D_{F^*}(\mathbf{y}||\hat{\mathbf{y}}) &= F^*(\mathbf{y}) - F^*(\hat{\mathbf{y}}) - \mathbf{f}^*(\hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}}) \\ &= \mathbf{f}^{-1}(\mathbf{y})^T\mathbf{y} - F(\mathbf{f}^{-1}(\mathbf{y})) - \mathbf{f}^{-1}(\hat{\mathbf{y}})^T\hat{\mathbf{y}} + F(\mathbf{f}^{-1}(\hat{\mathbf{y}})) - \mathbf{f}^{-1}(\hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}}) \\ &= \mathbf{f}^{-1}(\mathbf{y})^T\mathbf{y} - F(\mathbf{f}^{-1}(\mathbf{y})) + F(\mathbf{f}^{-1}(\hat{\mathbf{y}})) - \mathbf{f}^{-1}(\hat{\mathbf{y}})^T\mathbf{y} \end{aligned}$$

Finally, recall that $\hat{\mathbf{y}} = \mathbf{f}(\hat{\mathbf{z}})$ and $\mathbf{y} = \mathbf{f}(\mathbf{z})$, giving us

$$\begin{aligned} D_{F^*}(\mathbf{y}||\hat{\mathbf{y}}) &= \\ &= \mathbf{f}^{-1}(\mathbf{y})^T\mathbf{y} - F(\mathbf{f}^{-1}(\mathbf{y})) + F(\mathbf{f}^{-1}(\hat{\mathbf{y}})) - \mathbf{f}^{-1}(\hat{\mathbf{y}})^T\mathbf{y} \\ &= \mathbf{f}^{-1}(\mathbf{f}(\mathbf{z}))^T\mathbf{f}(\mathbf{z}) - F(\mathbf{f}^{-1}(\mathbf{f}(\mathbf{z}))) + F(\mathbf{f}^{-1}(\mathbf{f}(\hat{\mathbf{z}}))) - \mathbf{f}^{-1}(\mathbf{f}(\hat{\mathbf{z}}))^T\mathbf{f}(\mathbf{z}) \\ &= \mathbf{z}^T\mathbf{f}(\mathbf{z}) - F(\mathbf{z}) + F(\hat{\mathbf{z}}) - \hat{\mathbf{z}}^T\mathbf{f}(\mathbf{z}) \\ &= F(\hat{\mathbf{z}}) - F(\mathbf{z}) - \mathbf{f}(\mathbf{z})^T(\hat{\mathbf{z}} - \mathbf{z}) \\ &= D_F(\hat{\mathbf{z}}||\mathbf{z}) \end{aligned}$$

■

Theorem 1 *Given rank n input $t \times n$ matrix, X, and rank k output $t \times k$ matrix, Y, with $t > n$ and $t > k$, there exist unique global minimizers, W^* and U^* for $L(XW, Y)$ and $R(X, YU)$ respectively:*

$$W^* = \operatorname*{argmin}_W D_F(XW||f^{-1}(Y)) \tag{28}$$

$$U^* = \operatorname*{argmin}_U D_{F^*}(YU||f(X)) \tag{29}$$

Moreover, W^* and U^* are related in the following way

$$X^T\mathbf{f}(XW^*) = \mathbf{f}^{-1}(YU^*)^TY \tag{30}$$

Proof: Let F be a strictly convex function with $\mathrm{Dom}(F) = \{XW : W \in \mathbb{R}^n\}$, with any full rank X (i.e., X such that $XW_1 \neq XW_2$ for $W_1 \neq W_2$). Then $G = F(X\cdot)$ has $\mathrm{Dom}(G) = \mathbb{R}^n$ (which is convex). For W_1, W_2 in $\mathrm{Dom}(G)$ such that $W_1 \neq W_2$

$$\begin{aligned} G(\lambda W_1 + (1-\lambda)W_2) &= F(X(\lambda W_1 + (1-\lambda)W_2)) \\ &= F(\lambda XW_1 + (1-\lambda)XW_2) \\ &< \lambda F(XW_1) + (1-\lambda)F(XW_2) \quad \text{because } F \text{ is strictly convex and } XW_1 \neq XW_2. \\ &= \lambda G(W_1) + (1-\lambda)G(W_2) \end{aligned}$$

Therefore, G is strictly convex. The optimization $\min_W G(W)$ therefore has a unique minimum. Notice that we can always linearize X, W and Y to make sure that we are working with vectors.

For the relation, since W^* and U^* are global minimizers of $L(XW, Y)$ and $R(X, YU)$, we know that the gradients

$$\frac{d}{dW}L(XW^*, Y) = X^T\left(\mathbf{f}(XW^*) - Y\right) = \mathbf{0} \quad (n \times k)$$
$$\frac{d}{dU}R(X, YU^*) = Y^T\left(\mathbf{f}^*(YU^*) - X\right) = \mathbf{0} \quad (k \times n)$$

giving

$$X^T\left(\mathbf{f}(XW^*) - Y\right) = (Y^T\left(\mathbf{f}^*(YU^*) - X\right))^T$$
$$X^T\mathbf{f}(XW^*) - X^TY = \mathbf{f}^*(YU)^TY - X^TY$$

$\Longrightarrow$

$$X^T\mathbf{f}(XW^*) = \mathbf{f}^*(YU^*)^TY$$

■

Theorem 11 *For any X_L, X_U, Y_L, U and transfer function, f, with resulting affine feature set, $\mathcal{Z}$, then for $R(X, YU) = D_{F^*}(YU||f(X))$*

$$E[R(X_L, Y_LU)/t_L] = E[R(X, Z^*U)/t_S] + \\ E[R(Z_L^*U, Y_LU)/t_L] \tag{31}$$

where $X = [X_L; X_U]$ and $Z^* = \operatorname*{argmin}_{Z\in\mathcal{Z}} D_{F^*}(ZU||f(X))$

Proof: From the Generalized Pythagoras Theorem, we know that

$$E[D_{F^*}(Y_LU||f(X_L))/t_L] = \\ E[D_{F^*}(Z_L^*U||f(X_L))/t_L] + E[D_{F^*}(Z_L^*U||Y_LU)/t_L]$$

Since

$$E[D_{F^*}(Z_L^*U||f(X_L))/t_L] = E[D_{F^*}(Z^*U||f(X))/t_S] \\ = E[R(X, Z^*U)/t_S]$$

we get the above result. ■

9.2 Algorithms for clustering

To obtain the simplifications used for our modified clustering algorithms, we provide the following lemmas.

Lemma 12 $D_{F^*}(YU||\mathbf{f}(X)) = D_F(X||\mathbf{f}^*(YU)) = D_F(X||Y\mathbf{f}^*(U))$

Proof: From Lemma 4, we know that $D_F(X||\mathbf{f}^*(YU)) = D_{F^*}(YU||\mathbf{f}(X))$. Now, since $Y \in \{0,1\}^{t\times k}$ and $Y\mathbf{1} = \mathbf{1}$, we can see that YU simply selects rows of U, i.e. if there is a one at position $1 \leq j \leq k$, then row j in U is selected. Therefore,

$$\mathbf{f}^*(YU) = Y\mathbf{f}^*(U)$$

and we conclude that $D_F(X||\mathbf{f}^*(YU)) = D_F(X||Y\mathbf{f}^*(U))$. We can now optimize over M for $D_F(X||YM)$. ■

Lemma 13 *For a given $Y \in \{0,1\}^{t\times k}$ with $Y\mathbf{1} = \mathbf{1}$ and class j with $X \in Dom(\mathbf{f})$,*

$$\frac{1}{\mathbf{1}^TY_{:j}} \sum_{i:Y_{ij}=1} X_{i:} = \operatorname*{argmin}_{M\in Dom(\mathbf{f})} \sum_{i:Y_{ij}=1} D_F(X_{i:}||M_{:j})$$

Proof: Let n_j be the number of instances with class j, $\mathbf{m} = M_{:j}$, $\bar{\mathbf{x}} = \frac{1}{n_j}\sum_{i:Y_{ij}=1} X_{i:}$, $(\mathbf{m}, \bar{\mathbf{x}} \in \mathbb{R}^{n\times 1})$ and $\bar{F} = \frac{1}{n_j}\sum_{i:Y_{ij}=1} F(X_{i:})$. Now to simply $\frac{1}{n_j}\sum_{i:Y_{ij}=1} D_F(X_{i:}||\mathbf{m})$

$$\begin{aligned}\frac{1}{n_j}\sum_{i:Y_{ij}=1} D_F(X_{i:}||\mathbf{m}) &= \frac{1}{n_j}\sum_{i:Y_{ij}=1} F(X_{i:}) - F(\mathbf{m}) - \mathbf{f}(\mathbf{m})^T(X_{i:} - \mathbf{m}) \\ &= \bar{F} - \frac{1}{n_j}\sum_{i:Y_{ij}=1} F(\mathbf{m}) - \mathbf{f}(\mathbf{m})^T \frac{1}{n_j}\sum_{i:Y_{ij}=1}(X_{i:} - \mathbf{m}) \\ &= \bar{F} - F(\mathbf{m}) - \mathbf{f}(\mathbf{m})^T(\bar{\mathbf{x}} - \mathbf{m})\end{aligned}$$

and by definition

$$D_F(\bar{\mathbf{x}}||\mathbf{m}) = F(\bar{\mathbf{x}}) - F(\mathbf{m}) - \mathbf{f}(\mathbf{m})^T(\bar{\mathbf{x}} - \mathbf{m})$$

$$\Longrightarrow$$

$$\frac{1}{n_j}\sum_{i:Y_{ij}=1} D_F(X_{i:}||\mathbf{m}) = \bar{F} - F(\bar{\mathbf{x}}) + D_F(\bar{\mathbf{x}}||\mathbf{m})$$

$$\Longrightarrow$$

$$\sum_{i:Y_{ij}=1} D_F(X_{i:}||\mathbf{m}) = n_j\bar{F} - n_jF(\bar{\mathbf{x}}) + n_jD_F(\bar{\mathbf{x}}||\mathbf{m})$$

$$\Longrightarrow$$

$$\min_{\mathbf{m}} \sum_{i:Y_{ij}=1} D_F(X_{i:}||\mathbf{m}) = \min_{\mathbf{m}} D_F(\bar{\mathbf{x}}||\mathbf{m})$$

Since the Bregman divergence is guaranteed to be greater than or equal to zero, the minimum value for $D_F(\bar{\mathbf{x}}||\mathbf{m})$ is zero, obtained by setting $\mathbf{m} = \bar{\mathbf{x}}$. Therefore, for each instance i, the optimal setting for the inner minimization of $M_{:j} = \frac{1}{n_j}\sum_{i:Y_{ij}=1} X_{i:}$.

Notice that the objective value therefore is $n_j\bar{F} - n_jF(\bar{x})$, which is always non-negative because F is strictly convex so:

$$F(\bar{\mathbf{x}}) = F\left(\sum_{i:Y_{ij}=1} \frac{X_{i:}}{n_j}\right) < \sum_{i:Y_{ij}=1} \frac{1}{n_j} F(X_{i:}) = \bar{F}$$

■

From Lemmas 12 and 13, we get the following simplifications for Bregman hard clustering with non-linear transfers (Equations (22) and (24) in the main paper).

$$\begin{aligned}&\min_{Z\in\mathcal{Z}} \min_{U\in Dom(f^*)} D_{F^*}(ZU||f(X)) = \\ &\qquad = \min_{Z\in\mathcal{Z}} \min_{M\in Dom(f)\subset\mathbb{R}^{n\times k}} D_{F^*}(X||ZM) \\ &\qquad = \min_{Z\in\mathcal{Z}} \min_{M\in Dom(f)} \sum_{j=1}^{k} \sum_{i:Z_{ij}=1} D_{F^*}(X_{i:}||YM_{:j}) \\ &\qquad = \min_{Z\in\mathcal{Z}} \sum_{j=1}^{k} \frac{1}{\mathbf{1}^T Z_{:j}} \sum_{i:Z_{ij}} X_{i:}\end{aligned}$$

For mixture model clustering using standard EM, the unsimplified optimization given by Banerjee et al. [2005], with $\mathcal{S} = \{Z \in [0,1] \mid Z\mathbf{1} = 1\}$

$$\min_{Z\in\mathcal{S}} \min_{U} \sum_{i=1}^{t}\sum_{j=1}^{k} -\log(P_{F^*}(X_i|U_j))Z_{ij} \tag{32}$$

$$= \min_{Z\in\mathcal{S}} \min_{U\in\mathrm{Dom}(f)} \sum_{i=1}^{t}\sum_{j=1}^{k} -\log\left(e^{-D_F(X_i||U_j)}\right) Z_{ij} \tag{33}$$

They simplify the M-step using similar arguments to those for hard clustering. We define a slightly different optimization, now optimizing for the transfer $M = f(U)$ and then illustrate that we simplify the M-step for non-linear transfers. Note that in our optimization we move the sum and probability scaling inside the $\log$; this does not change the optimum because $\log$ is monotonic and the probabilities are always greater than or equal to zero. Note that we also add a smoothness parameter ρ; as$\rho \to \infty$, the objective approaches the hard clustering objective.

$$\min_{Z \in \mathcal{S}} \min_{M \in \mathrm{Dom}(f)} -\sum_{i=1}^{t} \log \left(\sum_{j=1}^{k} e^{-\rho D_F(X_i || M_j)} Z_{ij} \right) = \tag{34}$$

$$= \min_{\mathbf{p} \geq \mathbf{0}, \mathbf{p}^T \mathbf{1} = 1} \min_{M \in \mathrm{Dom}(f)} -\sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} p_j e^{-\rho D_F(X_i || M_{j:})} \right) \tag{35}$$

Again, the inner minimization over M simplifies to an expectation

$$M_{:j} = \frac{1}{\mathbf{1}^T Y_{:j}} \sum_{i=1}^{n} Y_{ij} X_{i:} \tag{36}$$

and we get the updates shown in Algorithm 2. For more details, look at the simplifications in Banerjee et al [2005].

9.3 Pseudocode and transfer functions

Below we provide pseudocode for our semisupervised regression approach, in Algorithm 9.3, and our semisupervised classification approach, Algorithm 9.3. The classification algorithm uses similar tricks from the unsupervised clustering algorithms provided in the paper. The regression algorithm simply uses a smooth optimizer (like limited memory BFGS) to alternate between optimizing Z and U according to the objective provided in Equation 26.

Algorithm 3 RevSemiSupRegression($X_L, X_U, Y_L, D_F, \boldsymbol{\beta}$)

```
// $\boldsymbol{\beta}$ is a weighting on samples, e.g. $\boldsymbol{\beta} = [\mathbf{1}; \boldsymbol{\mu}]$
Initialize $Y_U$ and $U$
$X = [X_L; X_U],\ K = k(X, X),\ \alpha = 0.1$
err$(Y_U, U) = \beta D_{F^*}([Y_L; Y_U] || \mathbf{f}(K))$
while (change in err$(Y_U, U)$) $>$ tol do
    $U = \mathrm{argmin}_U\, err(Y_U, U)$
    $Y_U = \mathrm{argmin}_Z\, err(Z, U)$
end while
$Y = [Y_L; Y_U]$
$A^* = \mathrm{argmin}_A\, D_F(KA || f^{-1}(Y)) + \alpha \mathrm{tr}(AA^T K)$
```

Below are the potential functions, inverses, forward losses and reverse losses for the transfers used in the paper. To make the tables cleaner, sometimes we will refer to $\hat{\mathbf{x}} = \mathbf{f}^{-1}(\mathbf{y}U)$. We omit D_{F^*} because it is not used in the clustering algorithm and is long, making the table difficult to read.

Algorithm 4 RevSemiSupSoftCluster$(X_L, X_U, Y_L, D_F, \boldsymbol{\beta})$

```
Initialize $M$ (e.g. $k$ randomly selected rows from $X$)
$p = \mathbf{1}/k$
$Y_U = [\,]$
$X = [X_L; X_U]$
err$(M,p) = -\sum_i \log\left(\sum_j p_j \exp(-\rho\boldsymbol{\beta} D_F(X_{i:}||M_{:j}))\right)$
while (change in err$(M,p)$) $>$ tol do
  //Shift Breg. divergence with min to avoid underflow
  E-Step:
        $Y_U(i,j) = p_j \exp[-\rho\mu(D_F(X_U(i,:)||M_{:j}) - \min_j D_F(X_U(i,:)||M_{:j}))]$
        $Y_U(i,j) = Y_U(i,j)/\sum_j Y_U(i,j)$
        $Z = [Y_L; Y_U]$
  M-Step:
        $M = \text{diag}(Z^T\mathbf{1})Z^T X$
        $p = \frac{1}{t} Z^T \mathbf{1}$
end while
```

Table 3: Transfer functions with their inverses and potential functions.

	$f(x)$	$f^{-1}(y)$	$F(\mathbf{x})$	$F^*(\mathbf{y})$
IDENTITY	$\mathbf{x}$	$\mathbf{x}$	$\mathbf{x}^2/2$	$\mathbf{y}^2/2$
SIGMOID	$\sigma(x) = (1+e^{-\mathbf{x}})^{-1}$	$\ln(\mathbf{y}/(\mathbf{1}-\mathbf{y}))$	$\mathbf{1}^T\ln(\mathbf{1}+e^{-\mathbf{x}})$	$\mathbf{y}\ln(\mathbf{y}/(\mathbf{1}-\mathbf{y})) + \mathbf{1}\ln(\mathbf{1}-\mathbf{y})$
SOFTMAX	$\xi(x) = e^{\mathbf{x}}/\mathbf{1}^T e^{\mathbf{x}}$	$\ln(\mathbf{y}) - \ln(\mathbf{y}_k)\mathbf{1}$	$\ln(\mathbf{1}^T e^{\mathbf{x}})$	$[\ln(\mathbf{y}) - \ln(\mathbf{y}_k)\mathbf{1}]\mathbf{y} - \ln(\mathbf{1}^T(\mathbf{y}-\mathbf{y}_k\mathbf{1}))$
EXP	$e^{\mathbf{x}}$	$\ln(\mathbf{y})$	$\mathbf{1}^T e^{\mathbf{x}}$	$[\ln(\mathbf{y}) - \mathbf{1}]\mathbf{y}^T$
CUBE	$\mathbf{x}^3$	$\mathbf{x}^{1/3}$	$\mathbf{1}^T\mathbf{x}^4/4$	$\mathbf{y}^{1/3}\mathbf{y}^T - 0.25\mathbf{y}^{4/3}\mathbf{1}$

Table 4: Transfer functions with forward and reverse losses.

	$D_F(\mathbf{x}W \Vert f^*(\mathbf{y}))$	$D_{F^*}(\mathbf{y}U \Vert f(\mathbf{x}))$
IDENTITY	$(\mathbf{x}W - \mathbf{y})^2/2$	$(\mathbf{x} - \mathbf{y}U)^2/2$
SIGMOID	$\mathbf{y}\ln(\mathbf{y}/\sigma(\mathbf{x}W)) + (\mathbf{1}-\mathbf{y})\ln((\mathbf{1}-\mathbf{y})/(\mathbf{1}-\sigma(\mathbf{x}W)$	$\mathbf{x}((\mathbf{1} - e^{-\mathbf{x}})/(\mathbf{1} + e^{-\mathbf{x}}))^T - \hat{\mathbf{x}}((\mathbf{1} - e^{-\hat{\mathbf{x}}})/(\mathbf{1} + e^{-\hat{\mathbf{x}}}))^T + \mathbf{1}\ln((\mathbf{1}+e^{-\hat{\mathbf{x}}})/(\mathbf{1}+e^{-\mathbf{x}}))^T$
SOFTMAX	$\ln(e^{\mathbf{x}W}\mathbf{1}^T) - \ln(\mathbf{1}(\mathbf{y}-\mathbf{y}_k\mathbf{1})^T) - \mathbf{y}W^T\mathbf{x}^T + \mathbf{y}(\mathbf{y}-\mathbf{y}_k\mathbf{1})^T$	OMITTED
EXP	$\mathbf{1}^T e^{\mathbf{x}W} - \mathbf{y}W^T\mathbf{x}^T$	$[\ln(\mathbf{y}U) - \mathbf{x} - \mathbf{1}]U^T\mathbf{y}^T + e^{\mathbf{x}}$
CUBE	$((\mathbf{x}W)^4\mathbf{1}^T)/4 - \mathbf{y}W^T\mathbf{x}^T$	$(\mathbf{y}U)^{1/3}U^T\mathbf{y}^T - 0.25(\mathbf{y}U)^{4/3}\mathbf{1} - \mathbf{x}U^T\mathbf{y}^T$