Supplementary Material for "Maximally Informative Hierarchical Representations of High-Dimensional Data"

A Proof of Theorem 2.2

Theorem. Basic Decomposition of Information

If Y is a representation of X and we define,

$$TC_L(X;Y) \equiv \sum_{i=1}^{n} I(Y:X_i) - \sum_{j=1}^{m} I(Y_j:X),$$

then the following bound and decomposition holds.

$$TC(X) \geq TC(X;Y) = TC(Y) + TC_L(X;Y)$$

Proof. The first inequality trivially follows from Eq. 2 since we subtract a non-negative quantity (a KL divergence) from $TC(X)$. For the second equality, we begin by using the definition of $TC(X;Y)$, expanding the entropies in terms of their definitions as expectation values. We will use the symmetry of mutual information, $I(A:B) = I(B:A)$, and the identity $I(A:B) = \mathbb{E}_{A,B}\log(p(a|b)/p(a))$. By definition, the full joint probability distribution can be written as $p(x,y) = p(y|x)p(x) = \prod_j p(y_j|x)p(x)$.

$$\begin{aligned} I(X:Y) &= \mathbb{E}_{X,Y}\left[\log\frac{p(y|x)}{p(y)}\right] \\ &= \mathbb{E}_{X,Y}\left[\log\frac{\prod_{j=1}^m p(y_j)}{p(y)}\frac{\prod_{j=1}^m p(y_j|x)}{\prod_{j=1}^m p(y_j)}\right] \\ &= -TC(Y) + \sum_{j=1}^{m} I(Y_j:X) \end{aligned} \tag{19}$$

Replacing $I(X:Y)$ in Eq. 2 completes the proof. □

B Proof of Theorem 2.4

Theorem. Upper Bounds on $TC(X)$

If $Y^{1:r}$ is a hierarchical representation of X and we define $Y^0 \equiv X$, and additionally $m_r = 1$ and all variables are discrete, then,

$$TC(X) \leq TC(Y^1) + TC_L(X;Y^1) + \sum_{i=1}^{n} H(X_i|Y^1)$$

$$TC(X) \leq \sum_{k=1}^{r}\left(TC_L(Y^{k-1};Y^k) + \sum_{i=1}^{m_{k-1}} H(Y_i^{k-1}|Y^k)\right).$$

Proof. We begin by re-writing Eq. 4 as $TC(X) = TC(X|Y^1) + TC(Y^1) + TC_L(X;Y^1)$. Next, for discrete variables, $TC(X|Y^1) \leq \sum_i H(X_i|Y)$, giving the inequality in the first line. The next inequality follows from iteratively applying the first inequality as in the proof of Thm. 2.3. Because $m_r = 1$, we have $TC(Y^r) = 0$. □

C Derivation of Eqs. 9 and 10

We want to optimize the objective in Eq. 8.

$$\begin{aligned} &\max_{p(y|x)} \sum_{i=1}^{n} \alpha_i I(Y:X_i) - I(Y:X) \\ &\text{s.t.} \sum_y p(y|x) = 1 \end{aligned} \tag{20}$$

For simplicity, we consider only a single Y_j and drop the j index. Here we explicitly include the condition that the conditional probability distribution for Y should be normalized. We consider α to be a fixed constant in what follows.

We proceed using Lagrangian optimization. We introduce a Lagrange multiplier $\lambda(x)$ for each value of x to enforce the normalization constraint and then reduce the constrained optimization problem to the unconstrained optimization of the objective $\mathcal{L}$.

$$\begin{aligned} \mathcal{L} = \sum_{\bar{x},\bar{y}} p(\bar{x})p(\bar{y}|\bar{x})\big(\sum_i \alpha_i(\log p(\bar{y}|\bar{x}_i) - \log p(\bar{y})) \\ -(\log p(\bar{y}|\bar{x}) - \log p(\bar{y}))\big) \\ +\sum_{\bar{x}} \lambda(\bar{x})(\sum_{\bar{y}} p(\bar{y}|\bar{x}) - 1) \end{aligned}$$

Note that we are optimizing over $p(y|x)$ and so the marginals $p(y|x_i), p(y)$ are actually linear functions of $p(y|x)$. Next we take the functional derivatives with respect to $p(y|x)$ and set them equal to 0. We re-use a few identities. Unfortunately, δ on the left indicates a functional derivative while on the right it indicates a Kronecker delta.

$$\begin{aligned} \frac{\delta p(\bar{y}|\bar{x})}{\delta p(y|x)} &= \delta_{y,\bar{y}}\delta_{x,\bar{x}} \\ \frac{\delta p(\bar{y})}{\delta p(y|x)} &= \delta_{y,\bar{y}}p(x) \\ \frac{\delta p(\bar{y}|\bar{x}_i)}{\delta p(y|x)} &= \delta_{y,\bar{y}}\delta_{x_i,\bar{x}_i}p(x)/p(x_i) \end{aligned}$$

Taking the derivative and using these identities, we obtain the following.

$$\begin{aligned} \frac{\delta\mathcal{L}}{\delta p(y|x)} = \lambda(x) + \\ p(x)\log\frac{\prod_i (p(y|x_i)/p(y))^{\alpha_i}}{p(y|x)/p(y)} + \\ \sum_{\bar{x},\bar{y}} p(\bar{x})p(\bar{y}|\bar{x})\big(\sum_i \alpha_i(\frac{\delta_{y,\bar{y}}\delta_{x_i,\bar{x}_i}p(x)}{p(x_i)p(\bar{y}|\bar{x}_i)} \\ -\delta_{y,\bar{y}}p(x)/p(\bar{y})) \\ -(\frac{\delta_{y,\bar{y}}\delta_{x,\bar{x}}}{p(\bar{y}|\bar{x})} - \delta_{y,\bar{y}}p(x)/p(\bar{y}))\big) \end{aligned}$$

Performing the sums over $\bar{x}, \bar{y}$ leads to cancellation of the last three lines. Then we set the remaining quantity equal to zero.

$$\frac{\delta\mathcal{L}}{\delta p(y|x)} = \lambda(x) + p(x)\log\frac{\prod_i p(y|x_i)/p(y)}{p(y|x)/p(y)} = 0$$

This leads to the following condition in which we have absorbed constants like $\lambda(x)$ in to the partition function, $Z(x)$.

$$p(y|x) = \frac{1}{Z(x)} p(y) \prod_{i=1}^{n} \left(\frac{p(y|x_i)}{p(y)} \right)^{\alpha_i}$$

We recall that this is only a formal solution since the marginals themselves are defined in terms of $p(y|x)$.

$$p(y) = \sum_{\bar{x}} p(\bar{x}) p(y|\bar{x})$$

$$p(y|x_i) = \sum_{\bar{x}} p(y|\bar{x}) p(\bar{x}) \delta_{x_i,\bar{x}_i} / p(x_i)$$

If we have a sum over independent objectives like Eq. 15 for $j = 1, \ldots, m$, we just place subscripts appropriately. The partition constant, $Z_j(x)$ can be easily calculated by summing over just $|Y_j|$ terms.

D Updates Do Not Decrease the Objective

The detailed proof of this largely follows the convergence proof for the iterative updating of the information bottleneck [3].

Theorem D.1. *Assuming $\alpha_1, \ldots, \alpha_n \in [0, 1]$, iterating over the update equations given by Eq. 11 and Eq. 10 never decreases the value of the objective in Eq. 8 and is guaranteed to converge to a stationary fixed point.*

Proof. First, we define a functional of the objective with the marginals considered as separate arguments. $\mathcal{F}[p(x_i|y), p(y), p(y|x)] \equiv$

$$\sum_{x,y} p(x) p(y|x) \left(\sum_i \alpha_i \log \frac{p(x_i|y)}{p(x_i)} - \log \frac{p(y|x)}{p(y)} \right)$$

As long as $\alpha_i \leq 1$, this objective is upper bounded by $TC_L(X;Y)$ and Thm. 2.3 therefore guarantees that the objective is upper bounded by the constant $TC(X)$. Next, we show that optimizing over each argument separately leads to the update equations given. We skip re-calculation of terms appearing in Sec. C. Keep in mind that for each of these separate optimization problems, we should introduce a Lagrange multiplier to ensure normalization.

$$\frac{\delta \mathcal{F}}{\delta p(y)} = \lambda + \sum_{\bar{x}} p(y|\bar{x}) p(\bar{x}) / p(y)$$

$$\frac{\delta \mathcal{F}}{\delta p(x_i|y)} = \lambda_i + \sum_{\bar{x}} p(y|\bar{x}) p(\bar{x}) \alpha_i \delta_{\bar{x}_i, x_i} / p(x_i|y)$$

$$\frac{\delta \mathcal{F}}{\delta p(y|x)} = \lambda(x) + p(x) \left(\sum_i \alpha_i \log \frac{p(x_i|y)}{p(x_i)} - \log \frac{p(y|x)}{p(y)} - 1 \right)$$

Setting each of these equations equal to zero recovers the corresponding update equation. Therefore, each update corresponds to finding a local optimum. Next, note that the objective is (separately) concave in both $p(x_i|y)$ and $p(y)$, because log is concave. Furthermore, the terms including $p(y|x)$ correspond to the entropy $H(Y|X)$, which is concave. Therefore each update is guaranteed to increase the value of the objective (or leave it unchanged). Because the objective is upper bounded this process must converge (though only to a local optimum, not necessarily the global one). □

E Convergence for S&P 500 Data

Fig. E.1 shows the convergence of the lower bound on $TC(X)$ as we step through the iterative procedure in Sec. 3.2 to learn a representation for the finance data in Sec. 5. As in the synthetic example in Fig. 3(a), convergence occurs quickly. The iterative procedure starts with a random initial state. Fig. E.1 compares the convergence for 10 different random initializations. In practice, we can always use multiple restarts and pick the solution that gives the best lower bound.

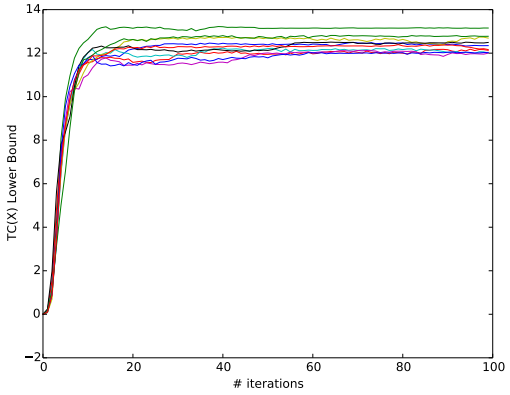

Figure E.1: Convergence of the lower bound on $TC(X)$ as we perform our iterative solution procedure, using multiple random initializations.

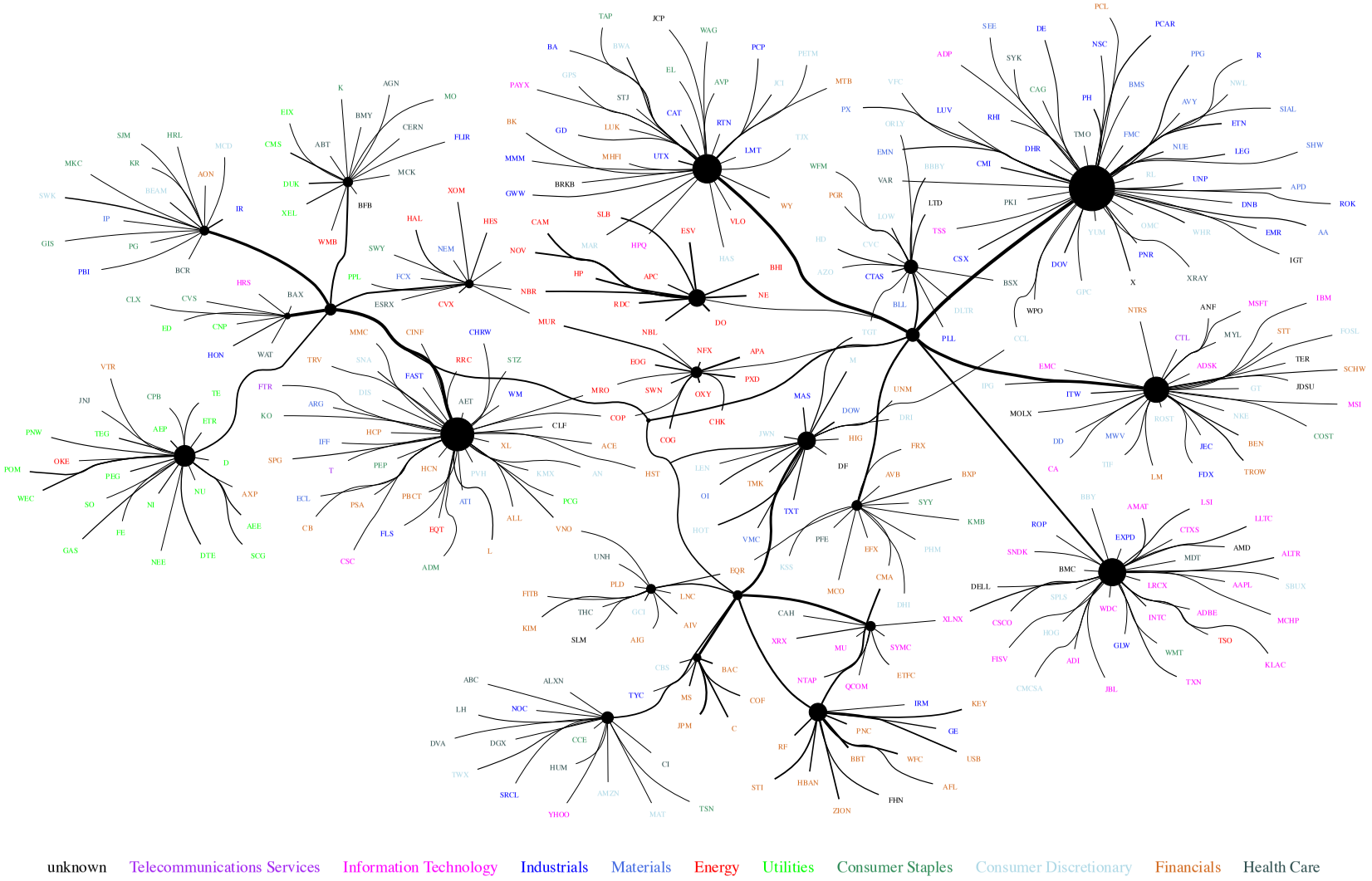

Figure E.2: A larger version of the graph in Fig. 4 with a lower threshold on for displaying edge weights (color online).

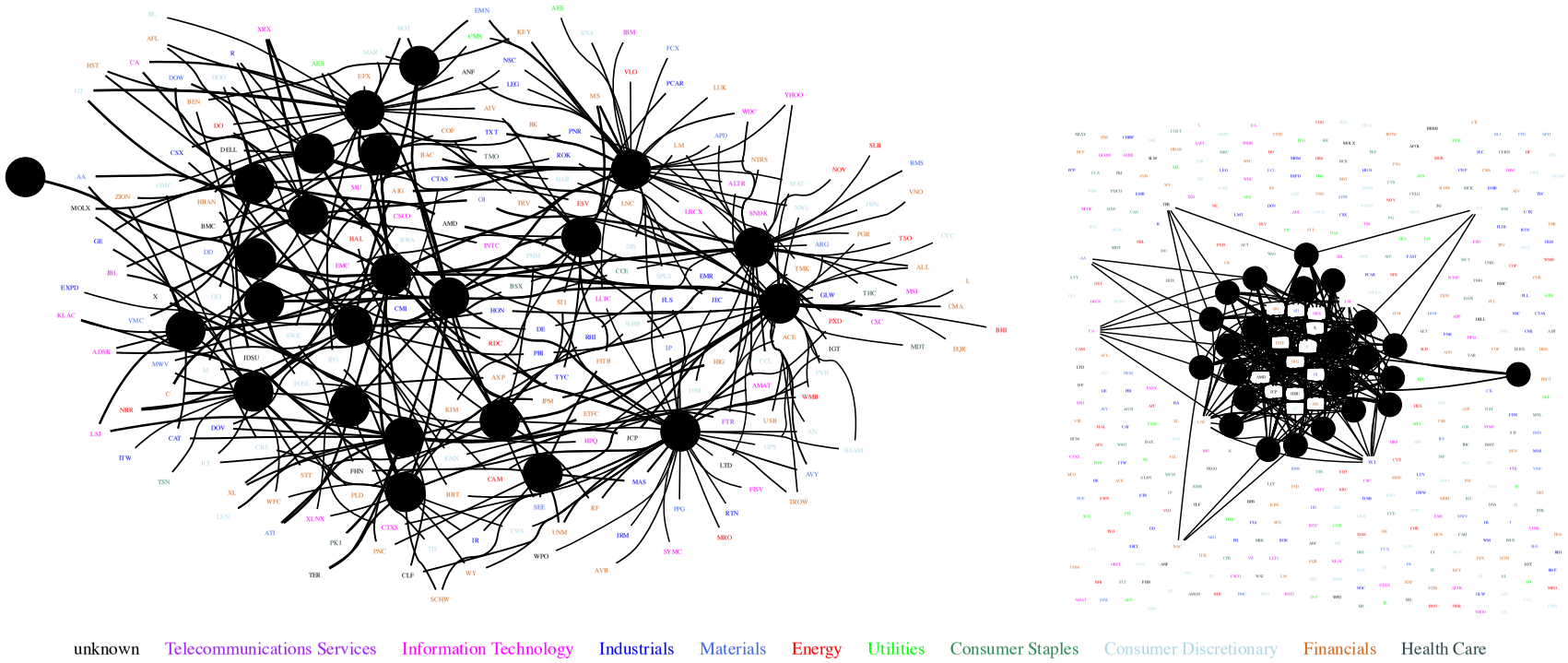

Figure E.3: For comparison, we constructed a structure similar to Fig. E.2 using restricted Boltzmann machines with the same number of layers and hidden units. On the right, we thresholded the (magnitude) of the edges between units to get the same number of edges (about 400). On the left, for each unit we kept the two connections with nodes in higher layers that had the highest magnitude and restricted nodes to have no more than 50 connections with lower layers (color online).