From Matrix Product States and Dynamical Mean-Field Theory to Machine Learning

Sommerfeld Theory Colloquium, LMU Munich
November 9, 2016

F. Alexander Wolf | falexwolf.de
Institute of Computational Biology
Helmholtz Zentrum München

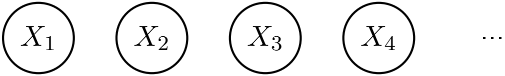
HelmholtzZentrum münchen
Deutsches Forschungszentrum für Gesundheit und Umwelt

HELMHOLTZ
| ASSOCIATION

Outline

- Matrix Product States / Tensor Trains
- Dynamical Mean-Field Theory
- Machine Learning

"Tensor Trains I": noninteracting bits

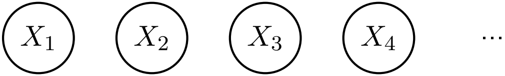


Vector of random variables $\boldsymbol{X} \in \{0,1\}^L$ with joint probability mass

$$p(\boldsymbol{x}) = \tfrac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = \sum_{n=1}^{L} x_n$$

normalized with $Z = \sum_{\boldsymbol{x}} e^{-H(\boldsymbol{x})/T}$.

"Tensor Trains I": noninteracting bits



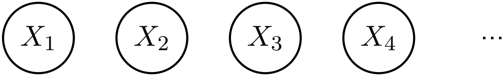
Vector of random variables $\boldsymbol{X} \in \{0,1\}^L$ with joint probability mass

$$p_{\boldsymbol{x}} = \tfrac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = \sum_{n=1}^{L} x_n$$

normalized with $Z = \sum_{\boldsymbol{x}} e^{-H(\boldsymbol{x})/T}$.

▷ $\boldsymbol{p}$ has 2^L components $\boldsymbol{x} \in \{(0,0,...,0),(0,0,...,1),\dots\}$.

“Tensor Trains I”: noninteracting bits



Vector of random variables $\boldsymbol{X} \in \{0,1\}^L$ with joint probability mass

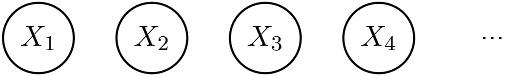
$$p_{\boldsymbol{x}} = \tfrac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = \sum_{n=1}^{L} x_n$$

normalized with $Z = \sum_{\boldsymbol{x}} e^{-H(\boldsymbol{x})/T}$.

▷ $\boldsymbol{p}$ has 2^L components $\boldsymbol{x} \in \{(0,0,...,0),(0,0,...,1),\ldots\}$.

▷ Note $2^{100} \simeq 10^{30} \simeq 10^{15}$ TB.

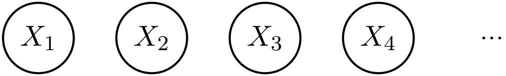
“Tensor Trains I”: noninteracting bits



Compute correlations via $\mathrm{cov}(X_n, X_m) = \langle X_n X_m \rangle - \langle X_n \rangle \langle X_n \rangle$,

$$\langle X_n X_m \rangle = \sum_{\boldsymbol{x}} x_n x_m \, p_{\boldsymbol{x}}.$$

“Tensor Trains I”: noninteracting bits



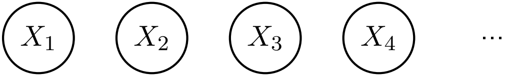
Compute correlations via $\mathrm{cov}(X_n, X_m) = \langle X_n X_m \rangle - \langle X_n \rangle \langle X_n \rangle$,

$$\langle X_n X_m \rangle = \sum_{\boldsymbol{x}} x_n x_m \, p_{\boldsymbol{x}}.$$

- ▷ Naive brute force: 2^L operations necessary.

- ▷ Monte Carlo: sampling in space of 2^L states.

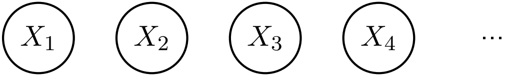
“Tensor Trains I”: noninteracting bits



Better: *independent* degrees of freedom X_n imply *separability*

$$
\begin{aligned}
p_{\boldsymbol{x}} = p_{x_1,x_2,\ldots,x_L} &= \frac{1}{Z} e^{-\sum_{n=1}^{L} x_n/T} \\
&= \frac{1}{Z} a_{x_1} a_{x_2} \ldots a_{x_L}, \quad a_{x_n} = e^{-x_n/T}.
\end{aligned}
$$

“Tensor Trains I”: noninteracting bits



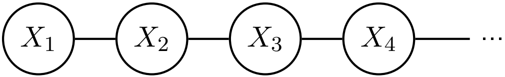
Better: *independent* degrees of freedom X_n imply *separability*

$$\begin{aligned} p_{\boldsymbol{x}} = p_{x_1,x_2,\ldots,x_L} &= \tfrac{1}{Z} e^{-\sum_{n=1}^{L} x_n/T} \\ &= \tfrac{1}{Z} a_{x_1} a_{x_2} \ldots a_{x_L}, \quad a_{x_n} = e^{-x_n/T}. \end{aligned}$$

Compute correlations in $2L$ operations ...

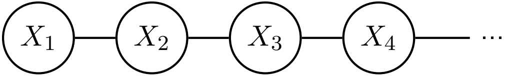
$$\begin{aligned} \langle X_n X_m \rangle &= \frac{1}{Z} \Big(\sum_{x_n} x_n a_{x_n} \Big) \Big(\sum_{x_m} x_m a_{x_m} \Big) \prod_{k \neq n,m}^{L} \Big(\sum_{x_k} a_{x_k} \Big) \\ &= \langle X_n \rangle \langle X_m \rangle \quad \ldots \quad \text{there are none.} \end{aligned}$$

"Tensor Trains II": interacting bits (Ising model)



$$\widetilde{\boldsymbol{p}}_{\boldsymbol{x}} = \tfrac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = -\sum_{n=1}^{L-1} x_n x_{n+1}.$$

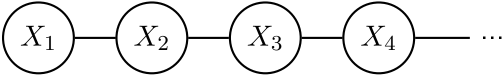
“Tensor Trains II”: interacting bits (Ising model)



Two-body interactions imply “almost – separability”

$$Z \sum_{\boldsymbol{x}} \widetilde{\boldsymbol{p}}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} e^{x_1 x_2/T} e^{x_2 x_3/T} \dots$$

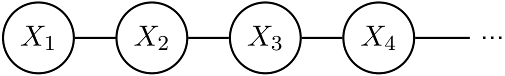
"Tensor Trains II": interacting bits (Ising model)



Two-body interactions imply "almost – separability"

$$Z \sum_{\boldsymbol{x}} \widetilde{\boldsymbol{p}}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} A_{x_1 x_2} A_{x_2, x_3} \dots$$

"Tensor Trains II": interacting bits (Ising model)



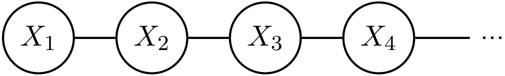
Two-body interactions imply "almost – separability"

$$Z \sum_{\boldsymbol{x}} \widetilde{\boldsymbol{p}}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} A_{x_1 x_2} A_{x_2, x_3} \dots$$

$$= \mathsf{gsum} AA \dots, \qquad A_{x_n x_{n+1}} = e^{x_n x_{n+1}/T}, \quad A \in \mathbb{R}^{2\times 2},$$

where gsum is the grand sum.

"Tensor Trains II": interacting bits (Ising model)



Two-body interactions imply "almost – separability"

$$Z \sum_{\boldsymbol{x}} \widetilde{\boldsymbol{p}}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} A_{x_1 x_2} A_{x_2, x_3} \ldots$$

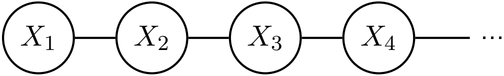
$$= \text{gsum} AA \ldots, \qquad A_{x_n x_{n+1}} = e^{x_n x_{n+1}/T}, \quad A \in \mathbb{R}^{2\times 2},$$

where gsum is the grand sum.

▷ Compare to non-interacting case

$$Z \sum_{\boldsymbol{x}} \boldsymbol{p}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} a_{x_1} a_{x_2} \ldots, \qquad a_{x_n} = e^{-x_n/T}, \quad a \in \mathbb{R}^2.$$

"Tensor Trains II": interacting bits (Ising model)

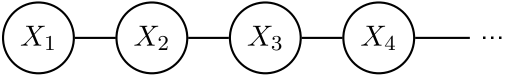


Compute correlations in $2^3 L$ operations (L matrix products)

$$\langle X_n X_m \rangle_{\widetilde{\boldsymbol{p}}} = \frac{1}{Z} \text{gsum} \prod_{k=1}^{n-1} \left(A^{[k]} \right) M \prod_{k=n}^{m-1} \left(A^{[k]} \right) M \prod_{k=m}^{L-1} \left(A^{[k]} \right)$$

$$\text{where} \quad M = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

“Tensor Trains II”: interacting bits (Ising model)



Compute correlations in $2^3 L$ operations (L matrix products)

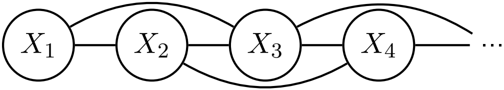
$$\langle X_n X_m \rangle_{\widetilde{\boldsymbol{p}}} = \frac{1}{Z} \mathsf{gsum} \prod_{k=1}^{n-1} \left(A^{[k]} \right) M \prod_{k=n}^{m-1} \left(A^{[k]} \right) M \prod_{k=m}^{L-1} \left(A^{[k]} \right)$$

$$\text{where} \quad M = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

▷ Compare to non-interacting case ($2L$ operations)

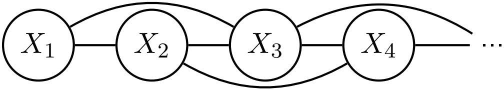
$$\langle X_n X_m \rangle_{\boldsymbol{p}} = \frac{1}{Z} \Big(\sum_{x_n} x_n A_{x_n} \Big) \Big(\sum_{x_m} x_m A_{x_m} \Big) \prod_{k \neq n,m} \Big(\sum_{x_k} A_{x_k} \Big)$$

“Tensor Trains III”: long-range interacting bit chain Wolf (2015)



$$p_{\boldsymbol{x}} = \frac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = -\sum_{n=1}^{L-2} x_n x_{n+1} x_{n+2}$$

"Tensor Trains III": long-range interacting bit chain Wolf (2015)



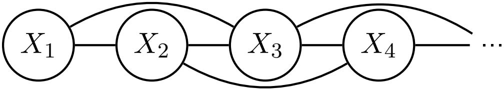
$$p_{\boldsymbol{x}} = \tfrac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = -\sum_{n=1}^{L-2} x_n x_{n+1} x_{n+2}$$

$$Z \sum_{\boldsymbol{x}} \boldsymbol{p}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} \prod_{n=1}^{L-2} A_{x_n x_{n+1} x_{n+2}}$$

$$A_{x_n x_{n+1} x_{n+2}} = e^{x_n x_{n+1} x_{n+2}/T}$$

$$A \in \mathbb{R}^{2\times 2\times 2}$$

"Tensor Trains III": long-range interacting bit chain Wolf (2015)



$$p_{\boldsymbol{x}} = \tfrac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = -\sum_{n=1}^{L-2} x_n x_{n+1} x_{n+2}$$

$$Z \sum_{\boldsymbol{x}} \boldsymbol{p}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} \prod_{n=1}^{L-2} A_{x_n x_{n+1} x_{n+2}}$$

$$A_{x_n x_{n+1} x_{n+2}} = e^{x_n x_{n+1} x_{n+2}/T}$$

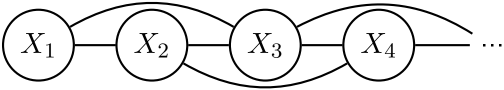
$$A \in \mathbb{R}^{2\times 2\times 2}$$

$$= \sum_{\boldsymbol{x}'} \prod_{n=1}^{L-2} B_{x'_n x'_{n+1}} B^t_{x'_{n+1} x'_{n+2}}$$

$$B_{x'_n (2x_{n+1}+x_{n+2})} = A_{x_n x_{n+1} x_{n+2}}$$

$$B \in \mathbb{R}^{2\times 4}$$

"Tensor Trains III": long-range interacting bit chain Wolf (2015)



$$p_{\boldsymbol{x}} = \frac{1}{Z} e^{-H(\boldsymbol{x})/T}, \quad H(\boldsymbol{x}) = -\sum_{n=1}^{L-2} x_n x_{n+1} x_{n+2}$$

$$Z \sum_{\boldsymbol{x}} \boldsymbol{p}_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} \prod_{n=1}^{L-2} A_{x_n x_{n+1} x_{n+2}} \qquad A_{x_n x_{n+1} x_{n+2}} = e^{x_n x_{n+1} x_{n+2}/T}, \quad A \in \mathbb{R}^{2\times 2\times 2}$$

$$= \sum_{\boldsymbol{x}'} \prod_{n=1}^{L-2} B_{x'_n x'_{n+1}} B^t_{x'_{n+1} x'_{n+2}} \qquad B_{x'_n (2x_{n+1}+x_{n+2})} = A_{x_n x_{n+1} x_{n+2}}, \quad B \in \mathbb{R}^{2\times 4}$$

Tensor Train format ▷ $\frac{1}{2}(2^3 + 4^3)L$ operations

“Tensor Trains” in Statistical Mechanics

- Write probability mass function

$$p : \{0, 1, ..., d\}^L \to \mathbb{R}, \quad d, L \in \mathbb{N},$$

as vector

$$p_{\boldsymbol{x}} = p(\boldsymbol{x}), \qquad \boldsymbol{p} \in \mathbb{R}^{d^L},$$

which is *indexed and parametrized* by $\boldsymbol{x} \in \{0, 1, ..., d\}^L$.

"Tensor Trains" in Statistical Mechanics

- Write probability mass function

$$p : \{0, 1, ..., d\}^L \to \mathbb{R}, \quad d, L \in \mathbb{N},$$

as vector

$$p_{\boldsymbol{x}} = p(\boldsymbol{x}), \qquad \boldsymbol{p} \in \mathbb{R}^{d^L},$$

which is *indexed and parametrized* by $\boldsymbol{x} \in \{0, 1, ..., d\}^L$.

If $p_{\boldsymbol{x}} = p(\boldsymbol{x})$ does *not* couple *all* index components x_n among each other, there is a low rank Tensor Train representation.

"Tensor Trains" in Statistical Mechanics

- Write probability mass function

$$p : \{0, 1, ..., d\}^L \to \mathbb{R}, \quad d, L \in \mathbb{N},$$

as vector

$$p_{\boldsymbol{x}} = p(\boldsymbol{x}), \qquad \boldsymbol{p} \in \mathbb{R}^{d^L},$$

which is *indexed and parametrized* by $\boldsymbol{x} \in \{0, 1, ..., d\}^L$.

If $p_{\boldsymbol{x}} = p(\boldsymbol{x})$ does *not* couple *all* index components x_n among each other, there is a low rank Tensor Train representation.

This reduces computational cost in summations over $p(\boldsymbol{x})$ from exponential to linear in system size.

"Tensor Trains" in Statistical Mechanics

- Write probability mass function

$$p : \{0, 1, ..., d\}^L \to \mathbb{R}, \quad d, L \in \mathbb{N},$$

  as vector

$$p_{\boldsymbol{x}} = p(\boldsymbol{x}), \qquad \boldsymbol{p} \in \mathbb{R}^{d^L},$$

  which is *indexed and parametrized* by $\boldsymbol{x} \in \{0, 1, ..., d\}^L$.

  If $p_{\boldsymbol{x}} = p(\boldsymbol{x})$ does *not* couple *all* index components x_n among each other, there is a low rank Tensor Train representation.

  This reduces computational cost in summations over $p(\boldsymbol{x})$ from exponential to linear in system size.

- What about quantum mechanics?

Statistical Mechanics vs. Quantum Mechanics

Instead of considering sums over classical weights, as in the partition sum,

$$1 = \sum_{\boldsymbol{x}} p_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} \langle \boldsymbol{x} | \hat{p}_{\boldsymbol{x}} | \boldsymbol{x} \rangle,$$

where we used a somewhat exaggerated notation.

Statistical Mechanics vs. Quantum Mechanics

Instead of considering sums over classical weights, as in the partition sum,

$$1 = \sum_{\boldsymbol{x}} p_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} \langle \boldsymbol{x} | \hat{p}_{\boldsymbol{x}} | \boldsymbol{x} \rangle,$$

where we used a somewhat exaggerated notation. We now consider quantum many-body states

$$|\psi\rangle = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}} |\boldsymbol{x}\rangle,$$

where $|\boldsymbol{x}\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_L\rangle = |x_1 x_2 \ldots x_L\rangle$ is a tensor product of single-particle basis states $|x_i\rangle$. For example

$$|x_i\rangle \in \{|\uparrow_i\rangle, |\downarrow_i\rangle\}$$

Statistical Mechanics vs. Quantum Mechanics

Instead of considering sums over classical weights, as in the partition sum,

$$1 = \sum_{\boldsymbol{x}} p_{\boldsymbol{x}} = \sum_{\boldsymbol{x}} \langle \boldsymbol{x} | \hat{p}_{\boldsymbol{x}} | \boldsymbol{x} \rangle,$$

where we used a somewhat exaggerated notation. We now consider quantum many-body states

$$|\psi\rangle = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}} |\boldsymbol{x}\rangle,$$

where $|\boldsymbol{x}\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_L\rangle = |x_1 x_2 \ldots x_L\rangle$ is a tensor product of single-particle basis states $|x_i\rangle$. For example

$$|x_i\rangle \in \{|\uparrow_i\rangle, |\downarrow_i\rangle\}$$

- But, do we know anything about how the vector of coefficients $\boldsymbol{c} = (c_{\boldsymbol{x}})$ *couples* its components, so that the tensor train format is meaningful?

For now we don't have to. Simply try an ansatz!

For now we don't have to. Simply try an ansatz!

- We can e.g. simply do a mean-field theory! Let us assume

$$c_{\boldsymbol{x}} \stackrel{!}{=} a^{x_1} a^{x_2} \ldots a^{x_L} = \prod_i a^{x_i}$$

then state can be manipulated doing $\sim L$ operations

$$|\psi\rangle = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}} |\boldsymbol{x}\rangle \stackrel{!}{=} |\psi_{\mathsf{MF}}\rangle = \sum_{\boldsymbol{x}} \prod_i a^{x_i} |\boldsymbol{x}\rangle = \prod_i^{\otimes} \Big(\sum_{x_i} a^{x_i} |x_i\rangle \Big)$$

For now we don't have to. Simply try an ansatz!

- We can e.g. simply do a mean-field theory! Let us assume

$$c_{\boldsymbol{x}} \stackrel{!}{=} a^{x_1} a^{x_2} \dots a^{x_L} = \prod_i a^{x_i}$$

  then state can be manipulated doing $\sim L$ operations

$$|\psi\rangle = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}} |\boldsymbol{x}\rangle \stackrel{!}{=} |\psi_{\mathsf{MF}}\rangle = \sum_{\boldsymbol{x}} \prod_i a^{x_i} |\boldsymbol{x}\rangle = \prod_i^{\otimes} \Big(\sum_{x_i} a^{x_i} |x_i\rangle \Big)$$

- How to determine the factors a^{x_i}? Variationally solve

$$\partial_{a^{x_i}} \frac{\langle \psi_{\mathsf{MF}} | H | \psi_{\mathsf{MF}} \rangle}{\langle \psi_{\mathsf{MF}} | \psi_{\mathsf{MF}} \rangle} = 0.$$

- Approximation to ground state. Approximation is *good* if ground state is in the same *class* of states as the ansatz $|\psi_{\mathsf{MF}}\rangle$.

Tensor Trains IV: Matrix Product States Schollwöck, arXiv:1008.3477 (2011)

- Relax mean-field assumption for coefficients of many body states

$$c_{\boldsymbol{x}} \stackrel{!}{=} a^{x_1} a^{x_2} a^{x_3} \ldots a^{x_L} = \prod_i a^{x_i}$$

to one that factorizes in matrices

$$c_{\boldsymbol{x}} \stackrel{!}{=} \sum_{\{\nu_i\}} A^{x_1}_{\nu_1} A^{x_2}_{\nu_1 \nu_2} A^{x_3}_{\nu_2 \nu_3} \ldots A^{x_L}_{\nu_L} = \prod_i A^{x_i}$$

Tensor Trains IV: Matrix Product States Schollwöck, arXiv:1008.3477 (2011)

- Relax mean-field assumption for coefficients of many body states

$$c_{\boldsymbol{x}} \stackrel{!}{=} a^{x_1} a^{x_2} a^{x_3} \dots a^{x_L} = \prod_i a^{x_i}$$

to one that factorizes in matrices

$$c_{\boldsymbol{x}} \stackrel{!}{=} \sum_{\{\nu_i\}} A^{x_1}_{\nu_1} A^{x_2}_{\nu_1 \nu_2} A^{x_3}_{\nu_2 \nu_3} \dots A^{x_L}_{\nu_L} = \prod_i A^{x_i}$$

- An MPS can be manipulated with costs of Lm^3, where m is the dimension of the matrices A^{x_i}

$$|\psi\rangle = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}} |\boldsymbol{x}\rangle \stackrel{!}{=} |\psi_{\mathsf{MPS}}\rangle = \sum_{\boldsymbol{x}} \prod_i A^{x_i} |\boldsymbol{x}\rangle$$

Tensor Trains IV: Matrix Product States Schollwöck, arXiv:1008.3477 (2011)

- Relax mean-field assumption for coefficients of many body states

$$c_{\boldsymbol{x}} \stackrel{!}{=} a^{x_1} a^{x_2} a^{x_3} \dots a^{x_L} = \prod_i a^{x_i}$$

to one that factorizes in matrices

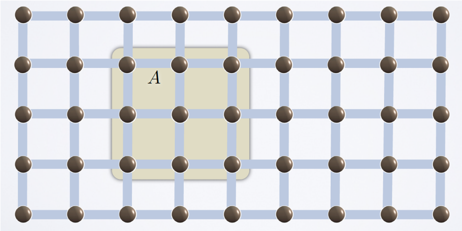
$$c_{\boldsymbol{x}} \stackrel{!}{=} \sum_{\{\nu_i\}} A^{x_1}_{\nu_1} A^{x_2}_{\nu_1 \nu_2} A^{x_3}_{\nu_2 \nu_3} \dots A^{x_L}_{\nu_L} = \prod_i A^{x_i}$$

- An MPS can be manipulated with costs of Lm^3, where m is the dimension of the matrices A^{x_i}

$$|\psi\rangle = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}} |\boldsymbol{x}\rangle \stackrel{!}{=} |\psi_{\text{MPS}}\rangle = \sum_{\boldsymbol{x}} \prod_i A^{x_i} |\boldsymbol{x}\rangle$$

- Are ground states in the same *class* as MPS? Which is this class? Are the coefficients $c_{\boldsymbol{x}}$ in ground states *weakly* coupled?

Tensor Trains IV: Weakly entangled states

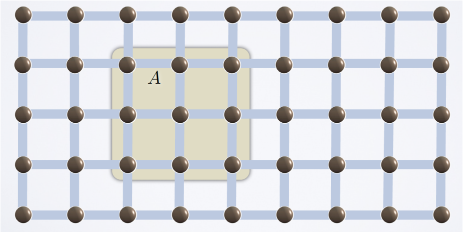


Eisert, arXiv:1308:3318 (2013)

Gapped Hamiltonians with short range interactions.

- Physical correlations have a finite range.
- Entanglement fulfills **area law**: entanglement of a region A is proportional to surface $|\partial A|$, not volume $|A|$, of this region.

Tensor Trains IV: Weakly entangled states



Eisert, arXiv:1308:3318 (2013)

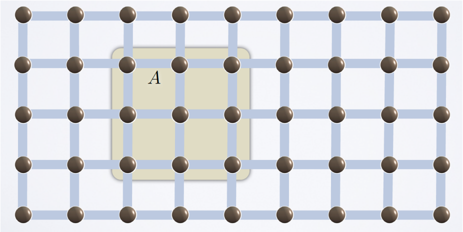
Gapped Hamiltonians with short range interactions.

- Physical correlations have a finite range.
- Entanglement fulfills **area law**: entanglement of a region A is proportional to surface $|\partial A|$, not volume $|A|$, of this region.

▷ There is a low-rank Tensor Train representation!

Dynamical Mean-Field Theory

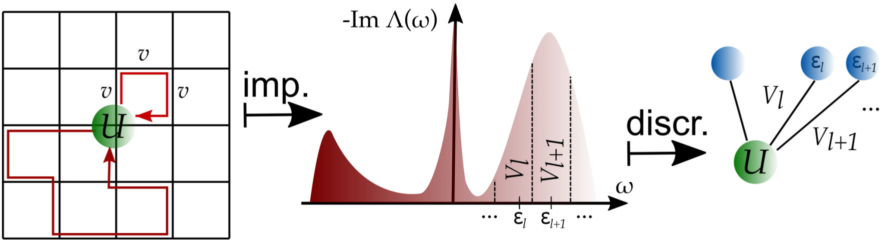
Quantum Embedding



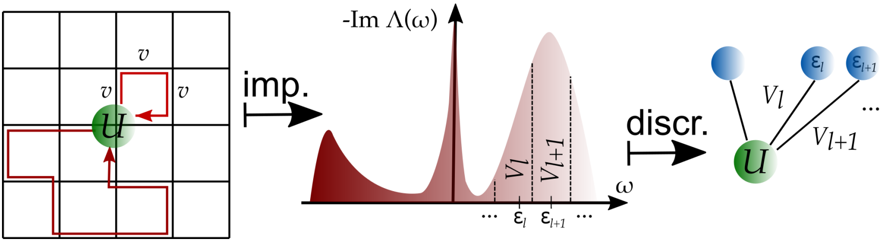
Eisert, arXiv:1308:3318 (2013)

- Dynamical Mean-Field Theory Metzner & Vollhardt (1989) Georges & Kotliar (1992)
- Density Matrix Embedding Theory Knizia & Chan, PRL 109, 186404 (2012)

Dynamical Mean-Field Theory

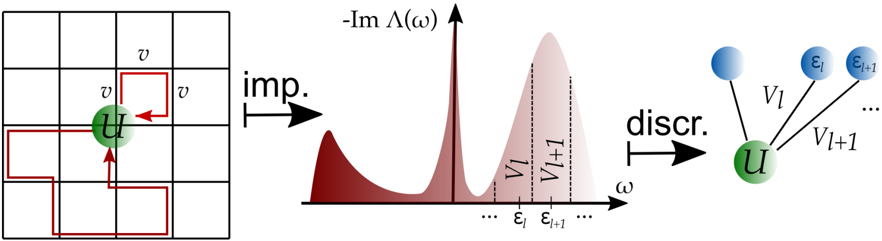


Dynamical Mean-Field Theory



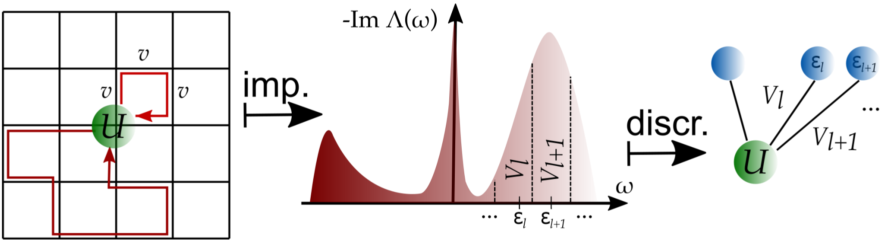
1. Find function $\Lambda(\omega)$ that describes the bath.

Dynamical Mean-Field Theory



1. Find function $\Lambda(\omega)$ that describes the bath.
2. Solve the reduced cluster problem.

Dynamical Mean-Field Theory



1. Find function $\Lambda(\omega)$ that describes the bath.
2. Solve the reduced cluster problem.

▷ Use Tensor Trains to represent the wave function of the cluster.

Tensor Trains and Dynamical Mean-Field Theory

Tensor Trains $\sim$ Density Matrix Renormalization Group (DMRG)

Algorithmic approaches

- Lanczos: unstable and imprecise

  García, Hallberg & Rozenberg, PRL 93, 246403 (2004)

Tensor Trains and Dynamical Mean-Field Theory

Tensor Trains $\sim$ Density Matrix Renormalization Group (DMRG)

Algorithmic approaches

- Lanczos: unstable and imprecise

  García, Hallberg & Rozenberg, PRL 93, 246403 (2004)

- dynamic (correction vector) DMRG: extremely expensive

  Nishimoto & Jeckelmann, J. Phys.: Cond. Mat. 16, 7063 (2004)

  Karski, Raas & Uhrig, PRB 72, 113110 (2005) Karski, Raas & Uhrig, PRB 77, 075116 (2008)

Tensor Trains and Dynamical Mean-Field Theory

Tensor Trains ~ Density Matrix Renormalization Group (DMRG)

Algorithmic approaches

- Lanczos: unstable and imprecise

  García, Hallberg & Rozenberg, PRL 93, 246403 (2004)

- dynamic (correction vector) DMRG: extremely expensive

  Nishimoto & Jeckelmann, J. Phys.: Cond. Mat. 16, 7063 (2004)

  Karski, Raas & Uhrig, PRB 72, 113110 (2005) Karski, Raas & Uhrig, PRB 77, 075116 (2008)

- Chebyshev and Fourier expansions: cheaper and precise

  Ganahl, Thunström, Verstraete, Held & Evertz, PRB 90, 045144 (2014)

  Wolf, McCulloch, Parcollet & Schollwöck, PRB 90, 115124 (2014a) ▷ 2-site cluster!

  Wolf, McCulloch & Schollwöck, PRB 90, 235131 (2014b) ▷ entanglement and non-EQ!

  Wolf, Justiniano, McCulloch & Schollwöck, PRB 91, 115144 (2015b) ▷ relation Chebyshev/ Fourier!

  de Vega, Schollwöck & Wolf, PRB 92, 155126 (2015) ▷ bath discretization!

Tensor Trains and Dynamical Mean-Field Theory

Tensor Trains $\sim$ Density Matrix Renormalization Group (DMRG)

Algorithmic approaches

- Lanczos: unstable and imprecise

  García, Hallberg & Rozenberg, PRL 93, 246403 (2004)

- dynamic (correction vector) DMRG: extremely expensive

  Nishimoto & Jeckelmann, J. Phys.: Cond. Mat. 16, 7063 (2004)

  Karski, Raas & Uhrig, PRB 72, 113110 (2005) Karski, Raas & Uhrig, PRB 77, 075116 (2008)

- Chebyshev and Fourier expansions: cheaper and precise

  Ganahl, Thunström, Verstraete, Held & Evertz, PRB 90, 045144 (2014)

  Wolf, McCulloch, Parcollet & Schollwöck, PRB 90, 115124 (2014a) ▷ 2-site cluster!

  Wolf, McCulloch & Schollwöck, PRB 90, 235131 (2014b) ▷ entanglement and non-EQ!

  Wolf, Justiniano, McCulloch & Schollwöck, PRB 91, 115144 (2015b) ▷ relation Chebyshev/ Fourier!

  de Vega, Schollwöck & Wolf, PRB 92, 155126 (2015) ▷ bath discretization!

- Imaginary axis: again cheaper!

  Wolf, Go, McCulloch, Millis & Schollwöck, PRX 5, 041032 (2015a) ▷ 2-site cluster for 3-band model!

Tensor Trains and Dynamical Mean-Field Theory

Tensor Trains $\sim$ Density Matrix Renormalization Group (DMRG)

Applications

- Non-thermal melting of Neel order in the Hubbard model

  Balzer, Wolf, McCulloch, Werner & Eckstein, PRX 5, 031039 (2015)

Tensor Trains and Dynamical Mean-Field Theory

Tensor Trains $\sim$ Density Matrix Renormalization Group (DMRG)

Applications

- Non-thermal melting of Neel order in the Hubbard model

  Balzer, Wolf, McCulloch, Werner & Eckstein, PRX 5, 031039 (2015)

- Benchmark quantum computing protocols

  Bauer, Wecker, Millis, Hastings & Troyer, PRX 6, 031045 (2016)

  Kreula, Clark & Jaksch, Sci. Rep. 6, 32940 (2016)

Tensor Trains and Dynamical Mean-Field Theory

Tensor Trains $\sim$ Density Matrix Renormalization Group (DMRG)

Applications

- Non-thermal melting of Neel order in the Hubbard model

  Balzer, Wolf, McCulloch, Werner & Eckstein, PRX 5, 031039 (2015)

- Benchmark quantum computing protocols

  Bauer, Wecker, Millis, Hastings & Troyer, PRX 6, 031045 (2016)

  Kreula, Clark & Jaksch, Sci. Rep. 6, 32940 (2016)

- In general: situations not treatable by QMC and NRG, which can be
  - correlated materials Linden *et al.*, in progress (2016)
  - gauge fields and topological phases

Machine Learning

Machine Learning

Estimate noisy functional relation

$$f : \mathcal{X} \to \mathcal{Y}, \qquad Y = f(X) + N,$$

from **data** $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n_{\text{samples}}}$.

Machine Learning

Estimate noisy functional relation

$$f : \mathcal{X} \to \mathcal{Y}, \qquad Y = f(X) + N,$$

from **data** $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n_{\text{samples}}}$.

- $f : \mathbb{R}^{28\times 28} \to \{2, 4\}$.

  Stoudenmire & Schwab, NIPS (2016)

Machine Learning

Estimate noisy functional relation

$$f : \mathcal{X} \to \mathcal{Y}, \qquad Y = f(X) + N,$$

from **data** $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n_{\text{samples}}}$.

- $f : \mathbb{R}^{28\times 28} \to \{2, 4\}$.

  Stoudenmire & Schwab, NIPS (2016)

- Linear regression using Gaussian noise model

$$p\big(y \big| x, \boldsymbol{\theta} = (\boldsymbol{w}, \sigma^2)\big) = \mathcal{N}\big(y \big| w_1 x + w_0, \sigma^2\big)$$

Machine Learning

Estimate noisy functional relation

$$f : \mathcal{X} \to \mathcal{Y}, \qquad Y = f(X) + N,$$

from **data** $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n_{\text{samples}}}$.

- $f : \mathbb{R}^{28\times 28} \to \{2, 4\}$.

  Stoudenmire & Schwab, NIPS (2016)

- Linear regression using Gaussian noise model

$$p\big(y\big|x, \boldsymbol{\theta} = (\boldsymbol{w}, \sigma^2)\big) = \mathcal{N}\big(y\big|w_1 x + w_0, \sigma^2\big)$$

Estimate parameters?

$$\boldsymbol{\theta}^* = \text{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{D}, \text{model}, \text{prior beliefs})$$

Machine Learning

Estimate noisy functional relation

$$f : \mathcal{X} \to \mathcal{Y}, \qquad Y = f(X) + N,$$

from **data** $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n_{\text{samples}}}$.

- $f : \mathbb{R}^{28\times 28} \to \{2, 4\}$.

  Stoudenmire & Schwab, NIPS (2016)

- Linear regression using Gaussian noise model

$$p\big(y\big|x, \boldsymbol{\theta} = (\boldsymbol{w}, \sigma^2)\big) = \mathcal{N}\big(y\big|w_1 x + w_0, \sigma^2\big)$$

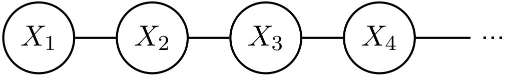
Estimate parameters?

$$\boldsymbol{\theta}^* = \text{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{D}, \text{model}, \text{prior beliefs})$$

▷ Integrate and optimize a high-dimensional distribution.

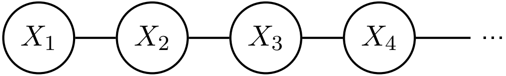
Graphical Models

Ising Model



Graphical Models

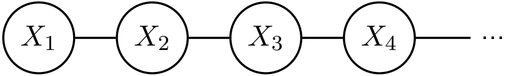
Ising Model



$$
\begin{aligned}
p(x_n) &= \sum_{\{x_{n'}\,|\,n' \neq n\}} p(x_1, ..., x_{n_{\mathsf{max}}}) \\
&= \sum_{\{x_{n'}\,|\,n' \neq n\}} A_{x_1 x_2} A_{x_2 x_3} \dots A_{x_{n_{\mathsf{max}}-1} x_{n_{\mathsf{max}}}}
\end{aligned}
$$

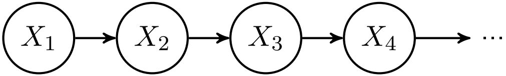
Graphical Models

Ising Model



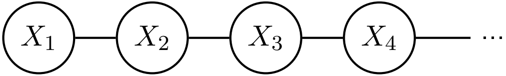
$$
\begin{aligned}
p(x_n) &= \sum_{\{x_{n'} \mid n' \neq n\}} p(x_1, ..., x_{n_{\mathsf{max}}}) \\
&= \sum_{\{x_{n'} \mid n' \neq n\}} A_{x_1 x_2} A_{x_2 x_3} \dots A_{x_{n_{\mathsf{max}}-1} x_{n_{\mathsf{max}}}}
\end{aligned}
$$

Markov Chain



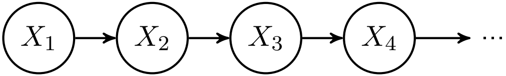
Graphical Models

Ising Model



$$
\begin{aligned}
p(x_n) &= \sum_{\{x_{n'} \mid n' \neq n\}} p(x_1, ..., x_{n_{\mathsf{max}}}) \\
&= \sum_{\{x_{n'} \mid n' \neq n\}} A_{x_1 x_2} A_{x_2 x_3} \ldots A_{x_{n_{\mathsf{max}}-1} x_{n_{\mathsf{max}}}}
\end{aligned}
$$

Markov Chain



$$
\begin{aligned}
p(x_n) &= \sum_{\{x_{n'} \mid n' \neq n\}} p(x_1, ..., x_{n_{\mathsf{max}}}) \\
&= \sum_{x_{n-1}} A_{x_n x_{n-1}} p(x_{n-1})
\end{aligned}
$$

▷ Here, the distribution itself factorizes!

Directed Acyclic Graphs

Markov chain

$$p(x_1, \ldots, x_{n_{\mathsf{max}}}) = p(x_1) \prod_{n=1}^{n_{\mathsf{max}}-1} p(x_{n+1}|x_n)$$

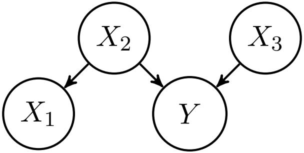
Directed Acyclic Graphs

Markov chain

$$p(x_1, \ldots, x_{n_{\mathsf{max}}}) = p(x_1) \prod_{n=1}^{n_{\mathsf{max}}-1} p(x_{n+1}|x_n)$$

General graph

$$p(x_1, \ldots, x_{n_{\mathsf{max}}}) = \prod_{n=1}^{n_{\mathsf{max}}} p(x_n|\mathsf{pa}(x_n))$$



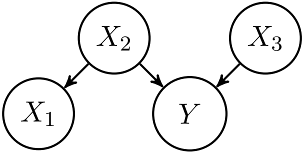
Directed Acyclic Graphs

Markov chain

$$p(x_1, \ldots, x_{n_{\mathsf{max}}}) = p(x_1) \prod_{n=1}^{n_{\mathsf{max}}-1} p(x_{n+1}|x_n)$$

General graph

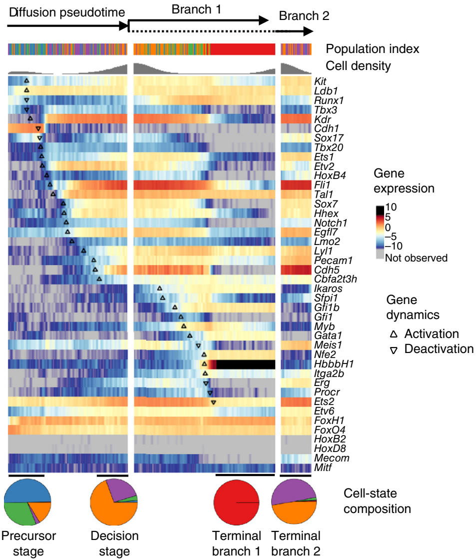
$$p(x_1, \ldots, x_{n_{\mathsf{max}}}) = \prod_{n=1}^{n_{\mathsf{max}}} p(x_n|\mathsf{pa}(x_n))$$



Example: $X_1 =$ yellow teeth, $X_2 =$ smoke, $Y =$ cancer, $X_3 =$ diet.

Inferring gene regulation from single-cell data

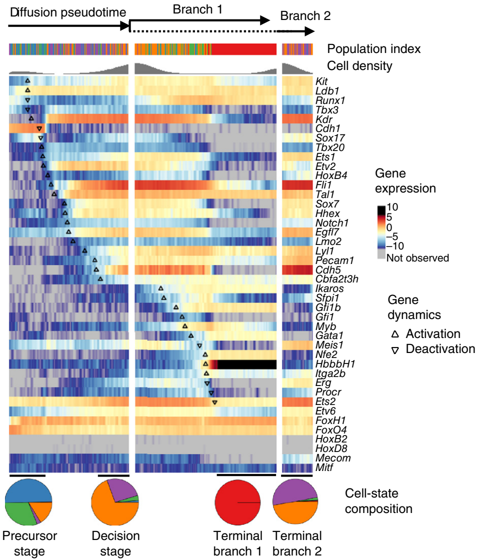
- Infer causal structure of gene regulation.



Haghverdi, Büttner, Wolf, Buettner & Theis, Nature Methods 13, 845 (2016)

Inferring gene regulation from single-cell data

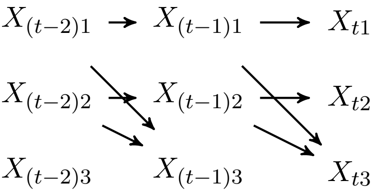
- Infer causal structure of gene regulation.
- Given a high-dimensional stochastic process, infer couplings among variables.



Haghverdi, Büttner, Wolf, Buettner & Theis, Nature Methods 13, 845 (2016)

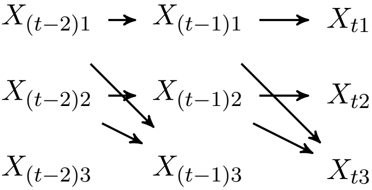
Time series data

Consider a d-dimensional time series $(\boldsymbol{X}_t)$, for example



Time series data

Consider a d-dimensional time series $(\boldsymbol{X}_t)$, for example



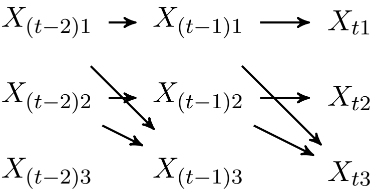
$$X_{t1} = X_{(t-1)1} + N_{t1}$$

$$X_{t2} = X_{(t-1)2} + N_{t2}$$

$$X_{t3} = X_{(t-1)1} \wedge \overline{X}_{(t-1)2} + N_{t3}$$

Time series data

Consider a d-dimensional time series $(\boldsymbol{X}_t)$, for example



$$X_{t1} = X_{(t-1)1} + N_{t1}$$

$$X_{t2} = X_{(t-1)2} + N_{t2}$$

$$X_{t3} = X_{(t-1)1} \wedge \overline{X}_{(t-1)2} + N_{t3}$$

One approach is **Transfer Entropy**, which is conditional mutual information Schreiber, PRL 85, 461 (2000) ($\sim$ Granger Causality Granger, Econometrica 37, 424 (1969))

$$\begin{aligned} \mathsf{TE}_{i\to j} &= \mathsf{MI}_{X_{(t-1)i};X_{tj}|S} \\ &= H_{X_{tj}|S} - H_{X_{tj}|X_{(t-1)i},S} \end{aligned}$$

where originally, $S = X_{(t-1)j}$, and later $S = \{$all observed variables$\}$.

Limitations of Transfer Entropy and Granger Causality

- Conditioning on all variables leads to terrible *curse of dimensionality*.

Limitations of Transfer Entropy and Granger Causality

- Conditioning on all variables leads to terrible *curse of dimensionality*.
- Say $X_1, X_2 \sim \text{Ber}(0.5)$, and $X_3 = X_1 + X_2$. Then X_3

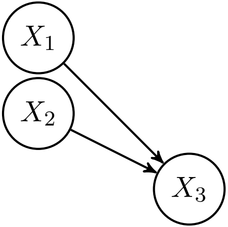
$$X_1 \not\perp\!\!\!\perp X_2 \mid X_3.$$

Limitations of Transfer Entropy and Granger Causality

- Conditioning on all variables leads to terrible *curse of dimensionality*.
- Say $X_1, X_2 \sim \text{Ber}(0.5)$, and $X_3 = X_1 + X_2$. Then X_3

$$X_1 \not\perp\!\!\!\perp X_2 \mid X_3.$$

  - ▷ Granger Causality and Transfer Entropy yield information flow $X_{(t-1)1} \to X_{t2}$. But it's non-causal, i.e. non-physical!

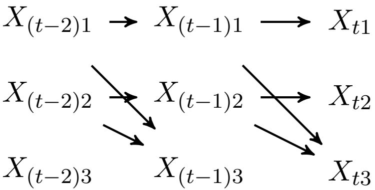


Limitations of Transfer Entropy and Granger Causality

- Conditioning on all variables leads to terrible *curse of dimensionality*.
- Say $X_1, X_2 \sim \text{Ber}(0.5)$, and $X_3 = X_1 + X_2$. Then X_3

$$X_1 \not\perp\!\!\!\perp X_2 \mid X_3.$$

  ▷ Granger Causality and Transfer Entropy yield information flow $X_{(t-1)1} \to X_{t2}$. But it's non-causal, i.e. non-physical!

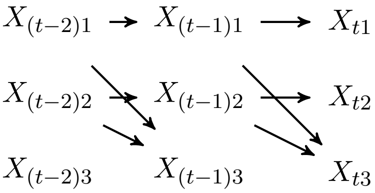


Limitations of Transfer Entropy and Granger Causality

- Conditioning on all variables leads to terrible *curse of dimensionality*.
- Say $X_1, X_2 \sim \text{Ber}(0.5)$, and $X_3 = X_1 + X_2$. Then X_3

$$X_1 \not\perp\!\!\!\perp X_2 \mid X_3.$$

  ▷ Granger Causality and Transfer Entropy yield information flow $X_{(t-1)1} \to X_{t2}$. But it's non-causal, i.e. non-physical!



▷ Need something different!

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.

SGS Test all combinations and conditions $X_i \perp\!\!\!\perp X_j | S$.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.

SGS Test all combinations and conditions $X_i \perp\!\!\!\perp X_j | S$.

PC(a) Test $X_i \perp\!\!\!\perp X_j | \varnothing$.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.

SGS Test all combinations and conditions $X_i \perp\!\!\!\perp X_j | S$.

PC(a) Test $X_i \perp\!\!\!\perp X_j | \varnothing$.

(b) On remaining edges and connected components, test $X_i \perp\!\!\!\perp X_j | X_k$.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.

SGS Test all combinations and conditions $X_i \perp\!\!\!\perp X_j | S$.

PC (a) Test $X_i \perp\!\!\!\perp X_j | \varnothing$.

(b) On remaining edges and connected components, test $X_i \perp\!\!\!\perp X_j | X_k$.

(c) And so forth.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.

   SGS Test all combinations and conditions $X_i \perp\!\!\!\perp X_j | S$.

   PC (a) Test $X_i \perp\!\!\!\perp X_j | \varnothing$.

   (b) On remaining edges and connected components, test $X_i \perp\!\!\!\perp X_j | X_k$.

   (c) And so forth.
3. Orient edges, where possible.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.
   - SGS Test all combinations and conditions $X_i \perp\!\!\!\perp X_j | S$.
   - PC(a) Test $X_i \perp\!\!\!\perp X_j | \varnothing$.
     - (b) On remaining edges and connected components, test $X_i \perp\!\!\!\perp X_j | X_k$.
     - (c) And so forth.
3. Orient edges, where possible.

- Doesn't work in gene expression time series as there is not enough dynamic noise.

Systematic conditional independence tests

Constraint based methods. Pearl & Verma (1991) Spirtes, Glymour & Scheines (2000)

1. Start with a fully connected graph.
2. Reduce edges by conditional independence tests.

SGS Test all combinations and conditions $X_i \perp\!\!\!\perp X_j | S$.

PC(a) Test $X_i \perp\!\!\!\perp X_j | \varnothing$.

(b) On remaining edges and connected components, test $X_i \perp\!\!\!\perp X_j | X_k$.

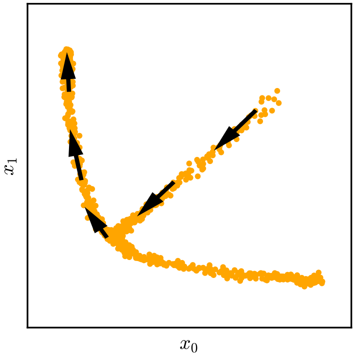
(c) And so forth.

3. Orient edges, where possible.

- Doesn't work in gene expression time series as there is not enough dynamic noise.

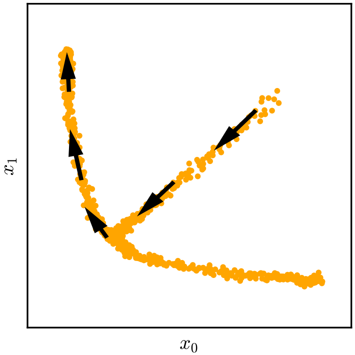
▷ In addition to *statistical association* among variables, test for *functional relation*. ▷ Geometry of data plays role. Wolf & Theis, in preparation (2016)

Integrating on the graph Wolf, Fischer & Theis, in preparation (2016)



$$\frac{dX_0}{dt} = \frac{X_0}{1+X_0}\frac{1}{1+X_1} - X_0 + N_0 =: V_0$$
$$\frac{dX_1}{dt} = \frac{X_1}{1+X_1}\frac{1}{1+X_0} - X_1 + N_1 =: V_1$$

Integrating on the graph Wolf, Fischer & Theis, in preparation (2016)



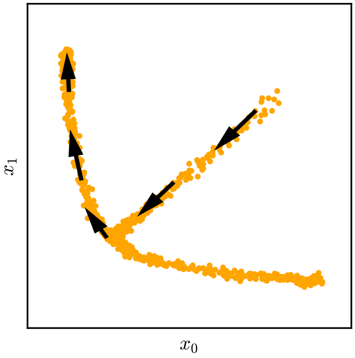
$$\frac{dX_0}{dt} = \frac{X_0}{1+X_0}\frac{1}{1+X_1} - X_0 + N_0 =: V_0$$

$$\frac{dX_1}{dt} = \frac{X_1}{1+X_1}\frac{1}{1+X_0} - X_1 + N_1 =: V_1$$

Statistical model $\widetilde{\boldsymbol{V}}$

$$\widetilde{V}_i = \sum_k \alpha_k X_k + \beta$$

Integrating on the graph Wolf, Fischer & Theis, in preparation (2016)



$$\frac{dX_0}{dt} = \frac{X_0}{1+X_0}\frac{1}{1+X_1} - X_0 + N_0 =: V_0$$

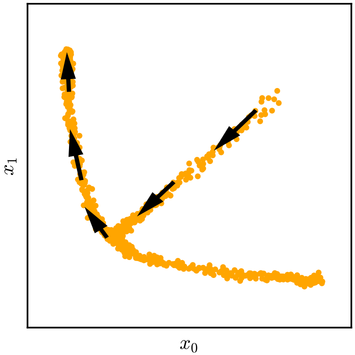
$$\frac{dX_1}{dt} = \frac{X_1}{1+X_1}\frac{1}{1+X_0} - X_1 + N_1 =: V_1$$

Statistical model $\widetilde{\boldsymbol{V}}$

$$\widetilde{V}_i = \sum_k \alpha_k X_k + \beta$$

Dynamics induced by $\widetilde{\boldsymbol{V}}$?

Integrating on the graph Wolf, Fischer & Theis, in preparation (2016)



$$\frac{dX_0}{dt} = \frac{X_0}{1+X_0}\frac{1}{1+X_1} - X_0 + N_0 =: V_0$$
$$\frac{dX_1}{dt} = \frac{X_1}{1+X_1}\frac{1}{1+X_0} - X_1 + N_1 =: V_1$$

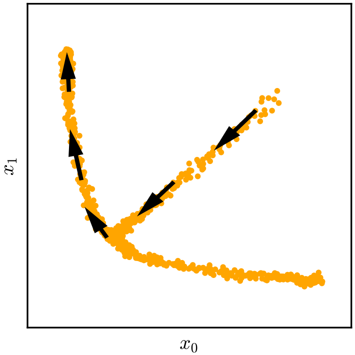
Statistical model $\widetilde{\boldsymbol{V}}$

$$\widetilde{V}_i = \sum_k \alpha_k X_k + \beta$$

Dynamics induced by $\widetilde{\boldsymbol{V}}$?

For the stochastic-mechanistic model, $\boldsymbol{X}(t) = \boldsymbol{X}_0 + \int_0^t dt\ \boldsymbol{V}(t)$.

Integrating on the graph Wolf, Fischer & Theis, in preparation (2016)



$$\frac{dX_0}{dt} = \frac{X_0}{1+X_0}\frac{1}{1+X_1} - X_0 + N_0 =: V_0$$
$$\frac{dX_1}{dt} = \frac{X_1}{1+X_1}\frac{1}{1+X_0} - X_1 + N_1 =: V_1$$

Statistical model $\widetilde{\boldsymbol{V}}$

$$\widetilde{V}_i = \sum_k \alpha_k X_k + \beta$$

Dynamics induced by $\widetilde{\boldsymbol{V}}$?

For the stochastic-mechanistic model, $\boldsymbol{X}(t) = \boldsymbol{X}_0 + \int_0^t dt\ \boldsymbol{V}(t)$.

For the statistic model $\widetilde{\boldsymbol{V}}$, “integrate on the graph”

$$A_{\boldsymbol{x}_i,\boldsymbol{x}_j} = \mathcal{N}\big(\boldsymbol{x}_i|\widetilde{\boldsymbol{x}}_i(\boldsymbol{x}_j),\sigma^2\big) \qquad \text{(Markov Model)}$$

Summary

- Tensor Trains/ Matrix Product States: low-rank factorization of high-dimensional distributions or wave functions.

Summary

- Tensor Trains/ Matrix Product States: low-rank factorization of high-dimensional distributions or wave functions.
- Dynamical Mean-Field Theory: learn something about a lattice problem from a single cluster.

Summary

- Tensor Trains/ Matrix Product States: low-rank factorization of high-dimensional distributions or wave functions.
- Dynamical Mean-Field Theory: learn something about a lattice problem from a single cluster.
- Graphical Models in Machine Learning: exact factorization of high-dimensional distribution wich applications, for example, in causal inference.

Summary

- Tensor Trains/ Matrix Product States: low-rank factorization of high-dimensional distributions or wave functions.
- Dynamical Mean-Field Theory: learn something about a lattice problem from a single cluster.
- Graphical Models in Machine Learning: exact factorization of high-dimensional distribution wich applications, for example, in causal inference.

Thanks to U. Schollwöck!

Summary

- Tensor Trains/ Matrix Product States: low-rank factorization of high-dimensional distributions or wave functions.
- Dynamical Mean-Field Theory: learn something about a lattice problem from a single cluster.
- Graphical Models in Machine Learning: exact factorization of high-dimensional distribution wich applications, for example, in causal inference.

Thanks to U. Schollwöck!

Thank you!

Balzer, K., F. A. Wolf, I. P. McCulloch, P. Werner & M. Eckstein, 2015, Phys. Rev. X **5**, 031039.

Bauer, B., D. Wecker, A. J. Millis, M. B. Hastings & M. Troyer, 2016, Phys. Rev. X **6**, 031045.

Eisert, J., 2013, Modeling and Simulation **3**, 520.

Ganahl, M., P. Thunström, F. Verstraete, K. Held & H. G. Evertz, 2014, Phys. Rev. B **90**, 045144.

García, D. J., K. Hallberg & M. J. Rozenberg, 2004, Phys. Rev. Lett. **93**, 246403.

Georges, A. & G. Kotliar, 1992, Phys. Rev. B **45**, 6479.

Granger, C. W. J., 1969, Econometrica **37**, 424.

Haghverdi, L., M. Büttner, F. A. Wolf, F. Buettner & F. J. Theis, 2016, Nature Methods **13**, 845.

Karski, M., C. Raas & G. S. Uhrig, 2005, Phys. Rev. B **72**, 113110.

Karski, M., C. Raas & G. S. Uhrig, 2008, Phys. Rev. B **77**, 075116.

Knizia, G. & G. K.-L. Chan, 2012, Phys. Rev. Lett. **109**, 186404.

Kreula, J. M., S. R. Clark & D. Jaksch, 2016, **6**, 32940, 1510.05703v3.

Metzner, W. & D. Vollhardt, 1989, Phys. Rev. Lett. **62**, 324.

Nishimoto, S. & E. Jeckelmann, 2004, J. Phys.: Condens. Matter **16**, 613.

Pearl, J. & T. Verma, 1991, A Theory of Inferred Causation, in Principles of Knowledge Representation and Reasoning: Proceeding of the Second International Conference, pp. 441–452.

Schollwöck, U., 2011, Annals of Physics **326**, 96.

Schreiber, T., 2000, Physical Review Letters **85**, 461.

Spirtes, P., C. Glymour & R. Scheines, 2000, Causation, Prediction, and Search (MIT Press, Cambridge, MA, USA), 2nd edition.

Stoudenmire, E. M. & D. J. Schwab, 2016, arXiv 1605.05775.

de Vega, I., U. Schollwöck & F. A. Wolf, 2015, Phys. Rev. B **92**, 155126.

Wolf, F. A., 2015, Solving dynamical mean-field theory using matrix product states, Ph.D. thesis, LMU Munich.

Wolf, F. A., D. S. Fischer & F. J. Theis, 2016, in preparation .

Wolf, F. A., A. Go, I. P. McCulloch, A. J. Millis & U. Schollwöck, 2015a, Phys. Rev. X **5**, 041032.

Wolf, F. A., J. A. Justiniano, I. P. McCulloch & U. Schollwöck, 2015b, Phys. Rev. B **91**, 115144.

Wolf, F. A., I. P. McCulloch, O. Parcollet & U. Schollwöck, 2014a, Phys. Rev. B **90**, 115124.

Wolf, F. A., I. P. McCulloch & U. Schollwöck, 2014b, Phys. Rev. B **90**, 235131.

Wolf, F. A. & F. J. Theis, 2016, to be submitted .