Coarse-graining and efficiently sampling with autoencoders

Gabriel STOLTZ
(CERMICS, Ecole des Ponts & MATHERIALS team, Inria Paris)

With Z. Belkacemi (Sanofi & ENPC), T. Lelièvre (ENPC/Inria) and P. Gkeka (Sanofi)

BRIN “Rare events” workshop, March 1st, 2023
Outline

• A (short/biased) review of machine learning approaches for CV

• Free-energy biasing and iterative learning with autoencoders\(^1\)
  • Autoencoders: definition, training, interpretation
  • Extended adaptive biasing force method
  • General presentation of the iterative algorithm
  • Illustration/sanity checks on toy examples

• Applications to systems of interest (alanine dipeptide, chignolin, HSP90)

• **ML reviews in MD** (biased towards dimensionality reduction, not force fields)

• **More general ML references**
    http://www.deeplearningbook.org
Molecular description of systems
What is the structure of the protein? What are its typical conformations, and what are the transition pathways from one conformation to another?
Statistical physics (2)

- **Microstate** of a classical system of $N$ particles:

$$ (q, p) = (q_1, \ldots, q_N, p_1, \ldots, p_N) \in \mathcal{E} = (a \mathbb{T})^{3N} \times \mathbb{R}^{3N} $$

Positions $q$ (configuration), momenta $p$ (to be thought of as $M \dot{q}$)

- **Hamiltonian** $H(q, p) = V(q) + \sum_{i=1}^{N} \frac{p_i^2}{2m_i}$ (physics is in $V$)

**Macrostate:** Boltzmann–Gibbs probability measure (NVT)

$$ \mu(dq \, dp) = Z_{NVT}^{-1} e^{-\beta H(q, p)} \, dq \, dp, \quad \beta = \frac{1}{k_B T} $$

- Typical evolution equations: Langevin dynamics (friction $\gamma > 0$)

$$ \begin{cases} 
 dq_t = M^{-1} p_t \, dt \\
 dp_t = -\nabla V(q_t) \, dt - \gamma M^{-1} p_t \, dt + \sqrt{2\gamma \beta^{-1}} \, dW_t
\end{cases} $$
Reaction coordinates (RC) / collective variables (CV)

- **Reaction coordinate** $\xi : (a\mathbb{T})^D \to \mathbb{R}^d$ with $d \ll D$

- Ideally: $\xi(q_t)$ captures the slow part of the dynamics

- **Free energy** computed on $\Sigma(z) = \{ q \in (a\mathbb{T})^D \mid \xi(q) = z \}$ (foliation)

$$F(z) = -\frac{1}{\beta} \ln \left( \int_{\Sigma(z)} e^{-\beta V(q)} \delta_{\xi(q)=z}(dq) \right)$$

- Various methods: TI, FEP, ABF, metadynamics, etc\(^2\)

---

\(^2\)Lelièvre/Rousset/Stoltz, *Free Energy Computations: A Mathematical Perspective* (Imperial College Press, 2010)
ML approaches for finding CV
Some representative approaches for finding CV (1)

- **Chemical/physical intuition** (distances, angles, RMSDs, coordination numbers, etc)

- **Short list of data-oriented approaches** (depending on the data at hand...)
  - [supervised learning] separate metastable states
  - [unsupervised/static] distinguish linear models (PCA) and nonlinear ones (e.g. based on autoencoders such as MESA³)
  - [unsupervised/dynamics] operator based approaches (VAC, EDMD, diffusion maps, MSM; incl. tICA and VAMPNets)

(Huge literature! I am not quoting precise references here because the list would be too long)

- **Other classifications⁴,⁵** possible, e.g. **slow vs. high variance CV**

---

Some representative approaches for finding CV (2)

### Methods for Choosing Collective variables

#### High-variance CVs
- Principal Components Analysis (PCA)
- Locally Linear Embedding (LLE)
- Independent Component Analysis (ICA)
- Laplacian and Hessian eigenmaps
- Local tangent space alignment
- Kernel PCA
- Nonlinear PCA
- Isomap
- Diffusion maps
- Multidimensional scaling
- Semidefinite embedding/
  Maximum variance unfolding

#### Available tools for CV identification
- Diffusion-Map-directed MD (DM-d-MD)
- Intrinsic Map Dynamics (iMapD)
- Smooth and nonlinear datadriven CVs (SandCV)
- Molecular Enhanced Sampling with Autoencoders (MESA)
- Reweighted Autoencoded Variations Bayes for Enhanced Sampling (RAVE)
- REinforcement Learning based on Adaptive sampling (REAP)

#### Slow CVs
- Variational Approach to Conformational dynamics (VAC)
- (extended) Dynamical Mode Decomposition ((E)DMD)
- Kernel TICA
- Markov State Models (MSM)
- Time-lagged autoencoders (TAEs)
- Time-lagged Independent Component Analysis (TICA)
- Deep Canonical Correlation Analysis (DCCA)
- Variational Dynamics Encoders (VDEs)
- Variational Approach for Markov Processes nets (VAMPnets)
- State-free Reversible VAMPnets (SRV)
CV construction with autoencoders
Bottleneck autoencoders (1)
Bottleneck autoencoders (2)

- Data space $\mathcal{X} \subseteq \mathbb{R}^D$, **bottleneck space** $\mathcal{A} \subseteq \mathbb{R}^d$ with $d < D$

$$f(x) = f_{\text{dec}}(f_{\text{enc}}(x))$$

where $f_{\text{enc}} : \mathcal{X} \rightarrow \mathcal{A}$ and $f_{\text{dec}} : \mathcal{A} \rightarrow \mathcal{X}$

Collective variable = encoder part

$$\xi = f_{\text{enc}}$$

- Fully connected neural network, symmetrical structure, $2L$ layers

- Parameters $\mathbf{p} = \{p_k\}_{k=1,...,K}$ (bias vectors $b_\ell$ and weights matrices $W_\ell$)

$$f_{\mathbf{p}}(x) = g_{2L} [b_{2L} + W_{2L} \cdots g_1 (b_1 + W_1 x)] ,$$

with activation functions $g_\ell$

(examples: $\tanh(x)$, ReLU $\max(0,x)$, sigmoid $\sigma(x) = 1/(1 + e^{-x})$, etc)
Training autoencoders

• Theoretically: minimization problem in $\mathcal{P} \subset \mathbb{R}^K$

$$p_\mu \in \arg\min_{p \in \mathcal{P}} \mathcal{L}(\mu, p),$$

with cost function

$$\mathcal{L}(\mu, p) = \mathbb{E}_\mu(\| X - f_p(X) \|^2) = \int_X \| x - f_p(x) \|^2 \mu(dx)$$

• In practice, access only to a sample: \textbf{minimization of empirical cost}

$$\mathcal{L}(\hat{\mu}, p) = \frac{1}{N} \sum_{i=1}^{N} \| x^i - f_p(x^i) \|^2, \quad \hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^i}$$

• Typical choices: \textit{canonical measure} $\mu$, data points $x^i$ postprocessed from positions $q$ (alignment to reference structure, centering, reduction to backbone carbon atoms, etc)
A variance interpretation of autoencoders

- **Total variance**
  \[
  \text{Var}(X) = \text{Var} \left[ \mathbb{E}(X | f_{\text{enc}}(X)) \right] + \mathbb{E} \left[ \text{Var}(X | f_{\text{enc}}(X)) \right]
  \]

- **Training w.r.t. decoder part** performed analytically in principle as
  \[
  \min_{f_{\text{enc}}} \left[ \min_{f_{\text{dec}}} \int_{\mathcal{X}} |x - f_{\text{dec}}(f_{\text{enc}}(x))|^2 \mu(dx) \right] = \min_{f_{\text{enc}}} \ell(f_{\text{enc}})
  \]
  with
  \[
  \ell(f_{\text{enc}}) = \mathbb{E} \left[ \left( X - \mathbb{E}(X | f_{\text{enc}}(X)) \right)^2 \right] = \mathbb{E} \left[ \text{Var}(X | f_{\text{enc}}(X)) \right] = \text{Var}(X) - \text{Var} \left[ \mathbb{E}(X | f_{\text{enc}}(X)) \right]
  \]

- **Minimizing** \( \ell(f_{\text{enc}}) \) equivalent to **maximizing** \( \text{Var} \left[ \mathbb{E}(X | f_{\text{enc}}(X)) \right] \)
  \( \Rightarrow \) minimizing intraclass dispersion vs. maximizing interclass dispersion
  \( \Rightarrow \) small spread of data points for \( f_{\text{enc}} \) given around the mean vs. the mean values associated with \( f_{\text{enc}} \) given should be as spread out as possible

- **Principal curve** interpretation, as for string method\(^6\)

---

\(^6\)Venturoli/Vanden–Eijnden (2009), Gerber/Whitaker (2013), Gerber (2021)
Some elements on training neural networks

- Many local minima...

- Actual procedure:
  - “Early stopping”: stop when validation loss no longer improves

- Choice of optimization method, here Adam
- No added regularization here (e.g. $\ell^1/\ell^2$, dropout, etc)

---

7 See Section 7.8 in [Goodfellow/Bengio/Courville]
8 See Chapter 8 in [Goodfellow/Bengio/Courville]
Free energy biasing
for complex CV
Extended systems

- Computing $\nabla \xi$ already difficult, higher order derivatives is worse

- **Extended system** strategy: $V_{\text{ext}}(q, \lambda) = V(q) + \frac{\kappa}{2}(\xi(q) - \lambda)^2$

- Free energy for the (simple) collective variable $\xi_{\text{ext}}(q, \lambda) = \lambda$

$$F_\kappa(z) = -\frac{1}{\beta} \ln \int_D e^{-\beta V_{\text{ext}}(q,z)} dq + C$$

$$= -\frac{1}{\beta} \ln \int \left( \int_{\Sigma(\zeta)} e^{-\beta V(q)} \delta_{\xi(q)-\zeta} dq \right) e^{-\beta \kappa (\zeta - z)^2 / 2} d\zeta + C$$

$$= -\frac{1}{\beta} \ln \int e^{-\beta F(\zeta)} \chi_\kappa(z - \zeta) d\zeta + \tilde{C}, \quad \chi_\kappa(s) = \left( \frac{\beta \kappa}{2\pi} \right)^{d/2} e^{-\beta \kappa s^2 / 2}$$

$$\kappa \to +\infty \quad F(z)$$

Calls for taking $\kappa$ large
Extended ABF

Extended overdamped Langevin dynamics \((\kappa \text{ limits } \Delta t\ldots)\)

\[
\begin{align*}
\begin{cases}
\frac{dq_t}{dt} & = \left[-\nabla V(q_t) + \kappa(\xi(q_t) - \lambda_t)\nabla \xi(q_t)\right] dt + \sqrt{2\beta^{-1}} dW_t^q \\
\frac{d\lambda_t}{dt} & = -\kappa[\lambda_t - \xi(q_t)] dt + \sqrt{2\beta^{-1}} dW_t^\lambda
\end{cases}
\end{align*}
\]

Bias by the free energy: add \(F'_\kappa(\lambda) = \text{steady state conditional average of } \kappa(\lambda - \xi(q))\)

Extended ABF overdamped Langevin dynamics

\[
\begin{align*}
\begin{cases}
\frac{dq_t}{dt} & = \left[-\nabla V(q_t) + \kappa(\xi(q_t) - \lambda_t)\nabla \xi(q_t)\right] dt + \sqrt{2\beta^{-1}} dW_t^q \\
\frac{d\lambda_t}{dt} & = \kappa[\xi(q_t) - \mathbb{E}(\xi(q_t) \mid \lambda_t)] dt + \sqrt{2\beta^{-1}} dW_t^\lambda
\end{cases}
\end{align*}
\]

In practice, \(\mathbb{E}(\xi(q_t) \mid \lambda_t)\) is estimated by

\[
\frac{\int_0^t \delta_\varepsilon(\lambda_s - \Lambda)\xi(q_s) \, ds}{\max\left(\eta, \int_0^t \delta_\varepsilon(\lambda_s - \Lambda) \, ds\right)}
\]
Unbiased estimate of the free energy in eABF

- **Stationarity**: configurations distributed according to

  \[ \rho(z, \lambda) = Z_\kappa^{-1} \exp \left( -\beta \left[ F(z) + \frac{\kappa}{2} (z - \lambda)^2 - F_\kappa(\lambda) \right] \right) \]

- **Unbiased estimator of the mean force (CZAR)**

  \[ F'(z) = -\frac{1}{\beta} \frac{d}{dz} \left[ \ln \bar{\rho}(z) \right] + \kappa (\langle \lambda \rangle_z - z) \]

  with \( \bar{\rho}(z) = \int \rho(z, \lambda) \, d\lambda \) and \( \langle \lambda \rangle_z = \frac{1}{\bar{\rho}(z)} \int \lambda \rho(z, \lambda) \, d\lambda \) (conditional dist.)

**Proof:** start from \( F'(z) = -\frac{1}{\beta} \frac{\partial_z \rho(z, \lambda)}{\rho(z, \lambda)} - \kappa (z - \lambda) \), multiply both sides of the equality by \( \rho(z, \lambda)/\bar{\rho}(z) \) and integrate with respect to \( \lambda \)

---

Joint distribution of $(\lambda, z)$ (deca-alanine)

logarithmic scale

$$\sigma^2 = \frac{1}{\beta \kappa}$$

Marginal distribution in $\lambda$ nearly uniform (as expected)
Iterative

free energy biasing/

toencoder learning
Training on modified target measures

- Interesting systems are **metastable** (no spontaneous exploration of phase space)
  Sample according to a biased distribution $\tilde{\mu}$ (**importance sampling**)

- Need for **reweighting**
  \[
  w(x) = \frac{\mu(x)}{\tilde{\mu}(x)}
  \]

- **Minimization problem**: theoretical cost function
  \[
  \mathcal{L}(\mu, p) = \int_X \| x - f_p(x) \|^2 w(x) \tilde{\mu}(dx),
  \]

  actual cost function
  \[
  \mathcal{L}(\hat{\mu}_{\text{wght}}, p) = \sum_{i=1}^N \hat{w}_i \| x^i - f_p(x^i) \|^2, \\
  \hat{w}_i = \frac{\mu(x^i)/\tilde{\mu}(x^i)}{\sum_{j=1}^N \mu(x^j)/\tilde{\mu}(x^j)}
  \]

- Only requires the knowledge of $\mu$ and $\tilde{\mu}$ up to a multiplicative constant.

- Minibatching: multinomial distribution for sampling with replacement

---

\[10\] As done in RAVE for instance, see Ribeiro/Bravo/Wang/Tiwary (2018), Wang/Ribeiro/Tiwary (2019)
Proof of concept with free energy biasing (1)

**Two dimensional potential** ("entropic switch")

\[
V(x_1, x_2) = 3e^{-x_1^2} \left( e^{-(x_2 - 1/3)^2} - e^{-(x_2 - 5/3)^2} \right) \\
- 5e^{-x_2^2} \left( e^{-(x_1 - 1)^2} + e^{-(x_1 + 1)^2} \right) + 0.2x_1^4 + 0.2(x_2 - 1/3)^4
\]

Trajectory from \( q^{j+1} = q^j - \nabla V(q^j) \Delta t + \sqrt{2\beta^{-1}} \Delta t G^j \) for \( \beta = 4 \) and \( \Delta t = 10^{-3} \) --- metastability in the \( x_1 \) direction

---

Proof of concept with free energy biasing (2)

- **Free energy biasing:** distributions $Z_i^{-1} \exp(-\beta [V(q) - F_i(\xi_i(q))])$

$$F_1(x_1) = -\frac{1}{\beta} \ln \left( \int_{\mathbb{R}} e^{-\beta V(x_1, x_2)} dx_2 \right), \quad F_2(x_2) = -\beta^{-1} \ln \left( \int_{\mathbb{R}} \ldots dx_1 \right)$$

**Three datasets:** unbiased trajectory, trajectories biased using $F_1$ and $F_2$

(free energy biased trajectories are shorter but same number of data points $N = 10^6$)

- Autoencoders: 2-1-2 topology, activation functions $\tanh$ (so that CV is in $[-1, 1]$) then identity

**Five training scenarios:**
- training on long unbiased trajectory (reference CV)
- $\xi_1$-biased trajectory, with or without reweighting
- $\xi_2$-biased trajectory, with or without reweighting
Proof of concept with free energy biasing (3)

Normalize to compare

\[ \xi_{\text{AE}}^\text{norm}(x) = \frac{\xi_{\text{AE}}(x) - \xi_{\text{AE}}^{\text{min}}}{\xi_{\text{AE}}^{\text{max}} - \xi_{\text{AE}}^{\text{min}}} \]

Reference CV (distinguishes well the 3 wells)

\( x_1 \)-biased trajectory

\( x_2 \)-biased trajectory
Full iterative algorithm  (Free Energy Biasing and Iterative Learning with AutoEncoders)

**Input:** Initial condition $q_0$, autoencoder topology and initialization parameters $A_{\text{init}}$, number of samples $N$, simulation procedure $S$ and adaptive biasing procedure $S_{\text{AB}}$, maximum number of iterations $I_{\text{max}}$, minimum convergence score $s_{\text{min}}$

**Initialization**
Sample traj$_0$ $\leftarrow$ $S(q_0, N)$
Initialize autoencoder $\text{AE}_0$ $\leftarrow$ $A_{\text{init}}$
Train $\text{AE}_0$ on traj$_0$ with weights $(\hat{w}_0, \ldots, \hat{w}_N) = (1, \ldots 1)$
Extract the encoder function $\xi_0 : x \mapsto \xi_0(x)$

**Iterative update of the collective variable**
Set $i \leftarrow 0$, $s \leftarrow 0$
**While** $i < I_{\text{max}}$ and $s < s_{\text{min}}$

- Set $i \leftarrow i + 1$
- Sample traj$_i$, $F_i$ $\leftarrow$ $S_{\text{AB}}(q_0, N, \xi_{i-1})$
- Compute weights $\hat{w}_j \propto e^{-\beta F_i(\xi_{i-1}(x^j))}$
- Initialize autoencoder $\text{AE}_i$ $\leftarrow$ $A_{\text{init}}$
- Train $\text{AE}_i$ on traj$_i$ with sample weights $\hat{w}_j$
- Extract the encoder function $\xi_i : x \mapsto \xi_i(x)$
- Set $s \leftarrow \text{regscore}(\xi_{i-1}, \xi_i)$

Set $\xi_{\text{final}} \leftarrow \xi_i$

**Production of output:**
Sample traj$_{\text{final}}$, $F_{\text{final}}$ $\leftarrow$ $S_{\text{AB}}(q_0, N_{\text{final}}\xi_{\text{final}})$ with $N_{\text{final}}$ large enough to ensure PMF convergence
Discussion on the convergence criterion (1/2)

• Check convergence of CV?
Quantify $\xi_i \approx \Phi(\xi_{i-1})$ for some monotonic function $\Phi$

• Approach: approximate $\Phi$ by a linear model (Nonlinear regression may be needed)

**Regression score** between $\xi$ and $\xi'$

- Two sets of values of CV $(\xi(q^1), \ldots, \xi(q^N))$ and $(\xi'(q^1), \ldots, \xi'(q^N))$
- Match them with a linear model $M(z) = Wz + b$

$$
\sum_{i=1}^{N} \| \xi'(q^i) - M(\xi(q^i)) \|^2
$$

- Coefficient of determination $R^2 = 1 - \frac{\sum_{i=1}^{N} \| \xi'(q^i) - \bar{\xi}' \|^2}{\sum_{i=1}^{N} \| \xi(q^i) - \bar{\xi} \|^2}$

- Maximization of $R^2$ w.r.t. $W, b$ provides $\text{regscore}(\xi', \xi)$

• Value of $s_{\text{min}}$ computed using some bootstrap procedure
Histogram of the $R^2$ scores obtained using subsets of $N = 10^5$ points out of $10^6$ points (vertical black line = 5% percentile).

(Left: Alanine dipeptide. Right: Chignolin)
The iterative algorithm on the toy 2D example

**Left:** with reweighting
Convergence to $\text{CV} \simeq x_1$

**Right:** without reweighting
No convergence
(cycles between two CVs)
Applications to systems of interest
Alanine dipeptide

- **Molecular dynamics:**
  openmm with openmm-plumed to link it with plumed
colvar module for eABF and computation of free energies\(^{12}\)
timestep 1 fs, friction \(\gamma = 1 \text{ ps}^{-1}\) in Langevin dynamics

- **Machine learning:**
  keras for autoencoder training
input = carbon backbone (realignement to reference structure and centering)
normal network: topology 24-40-2-40-24, tanh activation functions

\(^{12}\)See also Chen/Liu/Feng/Fu/Cai/Shao/Chipot, *J. Chem. Inf. Model.* (2022)
Ground truth computation

Long trajectory (1.5 $\mu$s), $N = 10^6$ (frames saved every 1.5 ps)

CV close to dihedral angles $\Phi, \Psi$

Quantify $s_{\text{min}} = 0.99$ for $N = 10^5$ using a bootstrapping procedure

For the iterative algorithm: 10 ns per iteration

(compromise between times not too short to allow for convergence of the free energy, and not too large in order to alleviate the computation cost)
Results for the iterative algorithm

<table>
<thead>
<tr>
<th>iter.</th>
<th>regscore</th>
<th>$(\Phi, \Psi)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>—</td>
<td>0.922</td>
</tr>
<tr>
<td>1</td>
<td>0.872</td>
<td>0.892</td>
</tr>
<tr>
<td>2</td>
<td>0.868</td>
<td>0.853</td>
</tr>
<tr>
<td>3</td>
<td>0.922</td>
<td>0.973</td>
</tr>
<tr>
<td>4</td>
<td>0.999</td>
<td>0.972</td>
</tr>
<tr>
<td>5</td>
<td>0.999</td>
<td>0.970</td>
</tr>
<tr>
<td>6</td>
<td>0.999</td>
<td>0.971</td>
</tr>
<tr>
<td>7</td>
<td>0.999</td>
<td>0.967</td>
</tr>
<tr>
<td>8</td>
<td>0.998</td>
<td>0.966</td>
</tr>
<tr>
<td>9</td>
<td>0.999</td>
<td>0.968</td>
</tr>
</tbody>
</table>
Chignolin (Folded/misfolded/unfolded states)
HSP90 (work in progress...)

Chaperone protein assisting other proteins to fold properly and stabilizing them against stress, including proteins required for tumor growth

→ look for inhibitors (e.g. targeting binding region of ATP; focus only on the N-terminal domain)

HSP90 (work in progress...)

6 conformational states, data from $10 \times 20$ ns trajectories, input features $= 207$ C carbons, AE topology 621-100-5-100-621

**Issue:** dimension of bottleneck?
Some perspectives
Some perspectives

- **Incorporating knowledge/information on the transition states?**
  - compute mean square error with respect to another distribution?
  - add terms to the loss function? (e.g. related to MEP\textsuperscript{13})

- **Incorporating dynamical information?**
  - time-lagged autoencoders and their variations\textsuperscript{14}
  - making use of the generator of the dynamics\textsuperscript{15}

- **Better understanding autoencoders**
  - choice of topology: mathematical analysis?
  - simple dimensionality reduction methods in the bottleneck (to allow for free energy computations)

\textsuperscript{13} Ramil/Boudier/Gorayeva/Marinica/Maillet, *J. Chem. Theory Comput.* (2022)
\textsuperscript{14} Chen/Sidky/Ferguson, *J. Chem. Phys.* (2019)
\textsuperscript{15} Zhang/Li/Schütte, *J. Comput. Phys.* (2022)
Interest in interfaces between ML and MD?

Program March-May 2024 at University of Chicago

Data-Driven Materials Informatics

Statistical Methods and Mathematical Analysis

March 4 — May 24, 2024

Add to Google Calendar

Description

Materials informatics is an emerging field defined by the use of simulation tools combined with methods from data sciences and machine learning to better understand materials properties and design innovative materials. The models which are considered cover an extremely wide range, from Schrödinger’s equation, which describes matter at the (sub)atomic scale, to the equations of continuum mechanics. Mathematical sciences play a key role in materials informatics, both to construct the databases used to train machine learning algorithms (since these databases are made of reference simulation results), and to harness them in order to extract the most relevant information. The aim of this program is bring together a diverse scientific audience, both between scientific fields (physical sciences, materials sciences, biophysics, etc) and within mathematics (mathematical modeling, numerical analysis, statistics and data analysis, etc), to make progress on key questions of materials informatics.