$$= \cancel{\mathbb{E}_D}\Big[\big(\underbrace{f(x_j) - \mathbb{E}_D[g(x_j;\hat{\theta}_D)]}_{}\big)^2\Big] +$$

bias^2

$$+ \mathbb{E}_D\Big[\big(g(x_j;\hat{\theta}_D) - \mathbb{E}_D[g(x_j;\hat{\theta}_D)]\big)^2\Big] + \text{cross } \cancel{\text{terms}} \to 0$$

because mean is equal to real mean.

variance

how similar is the model we learn every time to the average model you learn.

does not depend on the data on the training

$$\mathbb{E}[(C)] = \text{noise} + \text{bias}^2 + \text{variance}$$

noise $\hookrightarrow \sigma_\varepsilon^2$

how different is the true model f from the average model you learn.

blue graph is $C_{in}(\{y\}, \{g(\vec{x}, \theta)\}) = \sum_i (y_i - g(\vec{x}_i, \theta))^2$ → which gives you $\hat{\theta}_D$

red one is the Cost in $\mathbb{E}_{D,\varepsilon}[\,\text{Cost}(y, g(x, \hat{\theta}_D))]$

So what you care about is minimizing Cost and you need to remember about noise, bias^2 + variance.

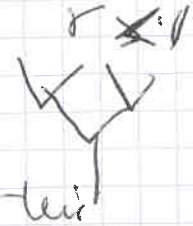
The exact definition of noise, bias^2 and variance only holds for Gaussian noise models but you get similar definitions for non-predictive cost functions.

Friday start here

Eleni lecture 4

So for optimization

$$\hat{C} = \sum \frac{Q_{el}}{C_e} + \lambda \sum C_e^{\gamma}$$



Now lets focus on how we get this and look at Development. (so growth).

lets look at the Murray model.

But make it look like energy.

$$E = \sum \frac{Q_e^2}{C_e} + C_0 \sum C_e^{\gamma}$$

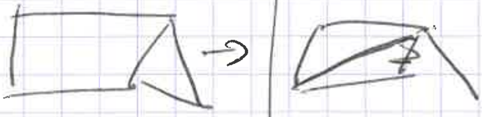
(same as previous eqn with notation changes)

If the network is developing $\frac{dC_e}{dt} \neq 0$ (growing)

$$\frac{dE}{dt} = -\sum \frac{Q_e^2}{C_e^2}\frac{dC_e}{dt} + C_0\gamma \sum C_e^{\gamma-1}\frac{dC_e}{dt} + \underbrace{\sum \frac{2Q_e}{C_e}\left(\nabla_C Q_e \cdot \frac{d\vec{C}}{dt}\right)}$$

this term much smaller

We do not consider topological network rearrangements, links are allowed to appear and disappear but not do this



These are called adaptive networks → a whole field

this term comes from the fact that Qe is a function of Ce.

$$= \sum\left(-\frac{Q_e^2}{C_e^2} + C_0\gamma C_e^{\gamma-1}\right)\boxed{\frac{dC_e}{dt}}$$

adaptation term

→ steady state $Q_e \sim C_e^{\frac{\gamma+1}{2}}$ motivates this

life can choose an adaptation ~~term~~ that ~~$\frac{dC_e}{dt}$~~ looks like:

$$\frac{dC_e}{dt} = +\frac{Q_e^2}{\gamma C_e^2} + C_0\gamma C_e^{\gamma-1}$$

use it

loose it

then we can write $\frac{dE}{dt}$ in this form. (because it the same thing as in the bracket above)

$$\frac{dE}{dt} = -\sum(\quad)^2$$

Sometimes (determined by the conductance) the system decides to use them a lot, then they grow, and they are used even more and so on → so an instability

Conversely, if you don't use it, it decreases.

General form of the adaptation term.

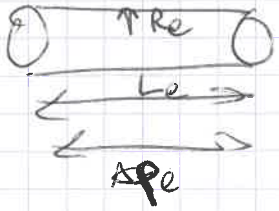
$$\frac{dC_e}{dt} = \beta Q_e^{g} - \alpha C_e \quad , \quad g = \frac{2}{\gamma+1}$$

in coupled flow systems? death rate tied to needs to vessels [illegible] (Poiseuille etc

this describes the architecture, the type of flow etc.

we transferred the Ce exponent to g (took it out), Remember Q_e is a function of C_e (it can be any function).

Go back to Poisset flow



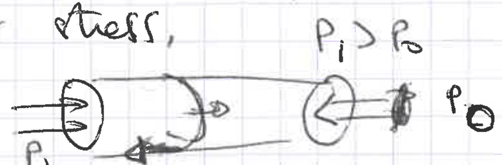
$$Q_e = \boxed{\frac{\pi R_e^4}{8 L \mu}} \Delta P_e$$

which means it pertains to this particular vessel.

"C_e — conductance

The vessels adapt by changing their diameter to increased shear stress.

Shear stress



$F = P_1 \pi R^2$

$F = P_0 \pi R^2$

$\Rightarrow \Delta F = \Delta P \pi R^2$

to balance the flow the wall exerts friction.

$\Delta F = \tau 2\pi R L$

shear stress

Force balance: $\Delta P \pi R^2 = 2\tau \pi R L \Rightarrow \boxed{\tau = \frac{4\mu Q}{\pi R^3}}$ shear stress

What people have measured (an eqn for the fact that vessels adapt to shear stress).

$$(*)\ \frac{dR_e}{dt} = \theta \left(|\tau_e|^\sigma - \tau_d^\sigma \right)$$

exponent that is a fitting parameter

parameter — actual value — desired value

$R_e \sim C_e^{1/4}$

$$\frac{dC_e}{dt} = \theta'' \left(\frac{|Q_e|^\sigma}{C_e^{3\sigma/4 - 1}} - \tau_d^\sigma C_e \right)$$

This agrees with Murray. Vessels optimize the cost and dissipation by optimizing (changing) shear stress.

(*) This is an exponential eqn. It means that the vessel mechanically would like to be receiving a specific shear stress. Because if its less you ~~wake~~ would recycle that vessel, waste of material. If its too much, it can burst. Vessels adapt, ~~but~~ it makes the decision to grow or decrease based on local ~~flow~~, local decision, local shear stress. This allows it to find a local minimum that affects the total system, because its hydrodynamically coupled. The vessel does not measure global flow, only local.

Question of timescales (these models assume all change is instantaneous), the other question is about spatial distance ⇒ how far away do you feel a perturbation ⇒ complicated question.

(If you ~~feel~~ measure the pressure in your ankle ~~get~~ can you figure out you had a stroke? ⇒ not right now with the current models).

Lets look at some limits:

$g \to 0$, $\gamma \to \infty$; steady state $C_e = \beta/\alpha$

large pipes are expensive — here you have loads of links and small links are used.

$g \gg 1$ $\gamma \to 0$; $\frac{dC_e}{dt} = \nu - \alpha C_e \Rightarrow$ ~~delet~~ exponential approach to $C_e = \nu/\alpha$

large pipes cheap — you get trees, you start deleting links below a certain treshold; every link used, much more economic design.

These optimisation eqn are also time optimisation eqns ⇒ they are solutions to MFPT eqns on graphs, and for random walks.

Topology and graph theory
(some generally applicable cool tools).

This also maps into other networks.

There is an exact mathematical analogy between spring networks and what we've seen being.

components in 1D but it generalises.

Flow	Electrical	Mechanical.
pressure p_i	voltage V_i	displacement x_i
current Q_{ij}	electrical current I_{ij}	spring force f_{ij} depends on the links
pressure drop Δp_i	voltage drop ΔU_i	elongation Δx (strain)
net current q_i	net current q_i	external forces or masses f_i
conductance C_{ij}	conductance C_i (conductor ( ))	spring constant k_{ij}
pressure drop flow relation $C\Delta p = Q$	Ohm's law $R=UI$	$F = k\Delta x$
current conservation $\Sigma Q_i = q$	current conservation	conservation of force $\Sigma F_i = 0$
dissipation $\frac{Q^2}{C} = \Delta p^2 \cdot C$		$k\Delta x^2 = E$ energy stored in a mechanical network.

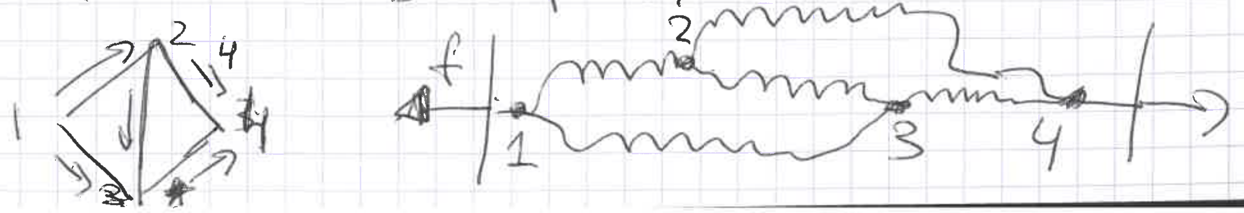
So we can take this code and measure the energy stored in a mechanical network etc.

So a current dipole becomes a mechanical dipole.

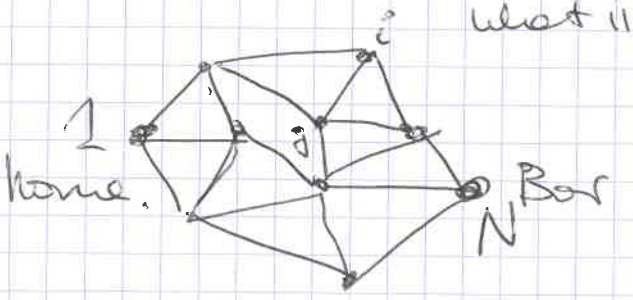
Engineers knew this in the 1960s.

(We still need computers.

→ They could not solve mechanical systems. They tested mechanical any network on electrical prototypes (because it was cheaper).



Random walks on graphs



what is the prob that a drunken sailor gets home before getting here.

$P_i^{(k)}$ = prob that a random walker is parked on node i at timestep k.

At each iteration of the random walk the walker will ~~[illegible]~~ move to a neighbouring node with prob $W_{ij} = \frac{a_{ij}}{\sum_j a_{ij}}$

Each link has a weight a_{ij} (how attractive the street is), normalized by the neighbours of that node.

(*)) $\vec{p}^{[k+1]} = \hat{D}^{-1}\hat{A}\vec{p}^{[k]}$

↑ prob of walker at each node

$\hat{D} = \text{diag}\{\sum_j a_{ij}\}$) sums all the links that are connected to node i

weighted adjacency matrix a_{ij}.

$\underbrace{(\hat{D}^{-1}\hat{A})}_{\overline{W}}$ will be related to the discrete laplacian → see next page

Boundary condition: if the walker starts at home, the prob ~~and~~ that you will be home at the end is 1, $P_1^\infty = 1$.

If you start at the bar, $P_N^\infty = 0$ you will never make it home.

So now we add boundary conditions.

If you know the P_i of all your neighbours, and you know P_{ij} then your P_i:

~~[illegible]~~ $P_i^\infty = \sum_j W_{ij} P_j^\infty$

is the weighted sum of all your neighbours.

$\Rightarrow \vec{P}^\infty = \hat{D}^{-1}\hat{A}\vec{p}^\infty$ ← so this is the steady mode of (*)

$$\hat{D}\vec{p}^\infty = \hat{A}\vec{p}^\infty \Rightarrow \underbrace{\left(\hat{D}-\hat{A}\right)}_{\hat{L} \leftarrow \text{discrete laplacian.}}\vec{p}^\infty = 0$$

$\vec{p}_i^\infty \leftarrow$ is the prob. that you end up at node i, before you get to N, so this really solved by the backward eqn. But A is symmetric so $\hat{W}^T = W$, (streets are bi'directional).

$$\hat{L}\vec{p}^\infty = 0$$

Travel time:

How long on average does it take to go from node i to to node 1 (expected # of steps).

If $t(v_i, 1) = t_i$

time from node i to node 1

If I knew ti that would be equal to the travel time from my neighbours + 1 (for one additional step).

$$\vec{t} = \underbrace{\hat{D}^{-1}\hat{A}}_{\hat{W}}\vec{t} + \vec{1}$$

this does not apply for node 1.

$\hat{W} \leftarrow$ size of $\hat{W}$ is $(N-1)\times(N-1)$ because we eliminate node 1.

$$\Rightarrow \underbrace{\left(\hat{D}-\hat{A}\right)}_{\hat{L}}\vec{t} = \underbrace{\hat{D}\vec{1}}_{-\vec{q}} \Rightarrow \boxed{\hat{L}\vec{t} = -\vec{q}}$$

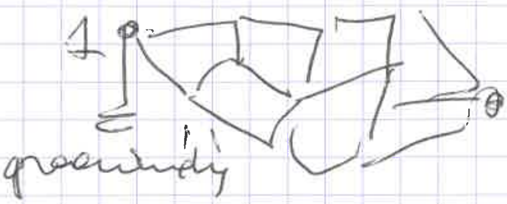
so this is an eqn. we've seen. But now we have sources.

Before we saw $L\vec{p} = -\vec{q}$.

But there is a subtlety now $\vec{q}(\vec{D})$ $\vec{q}$ depends on $\vec{D}$.

Before we had current BC, Now we fix the $t_1=0$

so in a leaf this would be grounding



But now the evapotranspiration depends on the neighbours! $\underline{q_i = -\sum a_{ij}}$

And this could be happening in the leaf!

Because things were strongly coupled.

Last thing

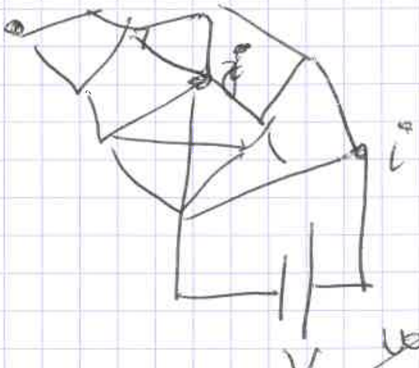
The travelling time from $1 \to i$ is not the same as $i \to 1$ (its easiest to get out of a small village and go to SF than from SF to a small village).

This translates into the need to symmetrize

$$\xi_{ij} = t(v_i, v_j) + t(v_j, v_i)$$

back and forth time (symmetrized time).

its the effective resistance of a graph.



$R_{eff} = \frac{\Delta P}{Q}$ $\left(\text{or } \frac{\Delta V}{Q} \right)$

$\boxed{\xi_{ij} = R_{eff}}$

voltage drop.

its the same thing as ~~the~~ effective resistance

it effectively gives you a **METRIC**

a distance measure!

Networks in all systems obey the same laws.

David lecture 2

gradient descent

$E(\theta) = \sum_{i=1}^{n} e_i(x_i, \theta)$ ← optimisation lecture.

$\vec{v}_t = \eta_t \nabla_\theta E(\theta_t)$ update rule.   step size.

$\theta_{t+1} = \theta_t - v_t$

actually works very well often.

If your steps are too small you ~~[illegible]~~ could they not go down because your step size is to big,

If its too small it will take forever,

Newton's method

$$E(\theta + v) \simeq E(\theta) + \nabla_\theta E(\theta)\, v + \frac{1}{2} v^T H v$$

$$0 = \nabla_\theta E(\theta) + H v_{opt}$$

$$v_t = H^{-1}(\theta_t)\, \nabla_\theta E(\theta_t)$$

$$\theta_{t+1} = \theta_t - v_t$$

maximal η is $\frac{2}{\lambda_{max}}$ (~~[illegible]~~ above that you explode)

a scheme for getting an idea for the ~~of~~ step size ~~whether you replace the Hessian.~~

you should use.

If you use different step sizes in different directions you will converge faster but it will not generalise as well,

You won't perform as well on the test data ~~for~~ although you will fit the train data better,

Stochastic gradient descent

$$\nabla_\theta E \to \sum_{i \in B_t} \nabla_\theta e_i(x_i, \theta)$$

you take a random subset of your data in each timestep B_t. In each time step its a different subset.

You can optimize the timestep and use different ones on top of that,

Stochastic GD (with momentum)

Whatever you pick your new v_t you have some memory of what happened at the previous step,

$v_t = \gamma v_{t-1} + \eta_t (\quad)$ so this is inertia.

How do you pick γ? In practice you pick a value, you use it, you test it, see how it works.

Next topic:

Bayesian inference.

Introduce a few concepts:

- likelihood function $p(x|\theta)$

  x is fixed, is a function of θ
- prior distribution $p(\theta)$
- posterior distribution $p(\theta|x) = \dfrac{p(x|\theta)\,p(\theta)}{\int d\theta'\, p(x|\theta')\,p(\theta')}$

  ⇒ now we are outputting a distribution of the parameters.

  You can then average over ~~this~~ distribution of parameters → proper Bayesian.

  This is often not done because it's complicated.

  So usually you use some frequency statistics,

$\langle\theta\rangle = \int d\theta\, \theta\, p(\theta|x)$ Bayes estimate

$\hat{\theta}_{MAP} = \underset{\theta}{\arg\max}\ p(\theta|x)$ maximum a posteriori estimate. → low variance, high bias.

If you want to work without a prior you can.

$\hat{\theta}_{ML} = \underset{\theta}{\arg\max}\ p(x|\theta)$ ML estimate.

↑ high variance, low bias regime.

Example: linear regression

$y = x^T w^* + \varepsilon$ ← true value.

likelihood $p(y|x,\theta) = N(y|\mu(x), \sigma^2)$

where $\mu = x^T w$ ← non true value.

$\theta = \{w, \sigma^2\}$

$\hat\theta = \arg\max_\theta \log p(y|x\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{M}(y_i - x_i^T w)^2 +$ const(σ)

choices for prior:

$p(\theta) = \prod_i \sqrt{\frac{\lambda}{2\pi}}\, e^{-\lambda\theta_i^2}$ Gaussian prior.

$p(\theta) = \prod_i \frac{\lambda}{2}\, e^{-\lambda|\theta_i|}$ ~~Exp penalty~~ exponential prior that results in an L_1 penalty.

$\hat\theta_{MAP} = \arg\max_\theta \log p(x|\theta)p(\theta) =$

Ridge regression $= \sum_{i=1} (y_i - x_i^T w)^2 + \underbrace{\lambda \|w\|_2^2}_{L_2 \text{ norm sense}}$

tries to get parameters small very, encourages sparsity

L_1 → lasso regression.

It pulls you away from fitting the wiggle of the data

⇒ regularization.

There is an equivalence between having a prior ~~and~~ having regularization (or penalizing parameter dispersion).

---

lets say you want to learn the Hamiltonian of an Ising model. (1D)

$$H = -J\sum_{j=1}^{L} S_j S_{j+1}$$

Learn J matrix. Our model does not know things are 1D, so we are going to sum over all pairs.

$H_{model} = \sum_{jk} J_{jk} S_j S_k$ so N^2 unknowns

we are going to do it with linear regression and then put on different ~~[illegible]~~ regularisation,

lasso does well, and it is the only one that gets the right model. ~~[illegible]~~ Others do badly.

And it would not have to be the case that we get the right model.

Here we get the right model because it's within the class of models we are considering.

But it does not have to be the case.

Next logistic regression (this is working on discrete noise).

→ discrimination.

logistic regression for classification,

The language of logistic regression is central for the language of neural networks.

logistic regression:

Our task is to figure out whether something is class 1 (dogs) or 0 (cats)

$y_i = 0$

$y_i = 1$

s takes features of the data & offset.

signal → $s = x^T w + b_0$

outputs zero or one:

PERCEPTRON $\sigma(s) = \text{sign}(s)$ ← this is what a perceptron does: sharp decision.

logistic regression does a soft version of this:

$\sigma(s) = \frac{1}{1+e^{-s}}$ LOGISTIC REGRESSION.

There a various ways of justifying $\frac{1}{1+e^{-s}}$ using hypothesis, but we will not do that,

$$p(y_i=1|x_i,\theta) = \frac{1}{1+e^{-(x_i^T w+b)}}$$

↑ prob of class 1 (dogs).

Write down likelihood model:

$$p(D|\theta) = \prod_{i=1}^{n} [\sigma(x_i^T, w)]^{y_i}[1-\sigma(x_i^T w)]^{1-y_i}$$

maximise the log-likelihood or the cost,

$$+\log p(D|\theta) = \Sigma + y_i \log \sigma(x_i^T w) + (1-y_i) \log (1-\sigma(x_i^T w))$$

this is called the "cross entropy"

To learn the model do gradient descent.
You can compute the gradient but you get a transcendental eqn, so you cannot set it to zero and get parameters, analytically. Can do it numerically.

Generalize this into not just binary outputs but more classes and ~~[illegible]~~ turn it into a neural network → next time.

Bill Bialek lecture (seminar)

RG-inspired to networks of real networks
The world is complicated. You cannot take into account everything that has been seen.
Why are you writing down simple models?
Two reasons: - the world really is simpler than it might be at the macroscopic' models
→ alike particles.

that is probably not the necessarily simple models made for biological systems: already molecules are complicated

— simplicity emerges as we try to describe macroscopic phenomena

→ phenomenology of place cell neurons.

RG

Let's think of systems that are in thermal equilibrium

If I have local variables $\{\sigma_i\}_{i=1,2,\dots N}$ N variables.

$$P(\{\sigma_i\}) = \frac{1}{Z} e^{-\beta(\{\sigma_i\})}$$

what does it mean to say that I am looking for simplicity to emerge as I am describing macroscopic phenomena.

original variables σ_i are on a scale a

let's imagine defining some coarse-grained variables on scale l $\tilde{\sigma}_i^{(l)}$ [$\tilde{N}$ variables]

(imagine defocusing your microscope).

what is the prob. distribution on scale l:

$$P_l(\{\sigma_i^l\}) = ?$$

$$E = g_1 \hat{O}(\{\sigma_i\}) + g_2 \hat{O}_2(\{\sigma_i\}) + \dots$$

I can expand the energy in some way.

I can think of the probability distribution that I end up with also as a Boltzmann distribution ~~[illegible]~~ with the energy as an expansion in the new σ_i^l basis. If still have N variables,

$$P_l(\{\tilde{\sigma}_i^l\}) = ? \sim e^{-\beta E_{(l)}(\{\sigma_i^l\})}$$

$$E_{(l)} = \tilde{g}_1^l \hat{O}(\{\tilde{\sigma}_i^l\}) + \tilde{g}_2^l \hat{O}(\{\sigma_i^l\}) + \dots$$

but now I need less coefficients.

I still have the same # of variables, but lots coefficients become most go to zero. That is what it means that the model is simpler.

$$(g_1, g_2 \ldots)_a \rightarrow (\tilde{g}_1, \tilde{g}_2, \ldots)_l$$

a lot goes in here

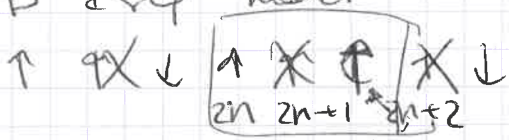
$\frac{d\tilde{g}_1}{dl} > 0$ relevant variable

$\frac{d\tilde{g}_n}{dl} < 0$ irrelevant

(everything under the rug all the RG flow to a fixed point)

if most terms were relevant and a few irrelevant we would be in trouble. Luckily it's the other way round: most terms are irrelevant.

Exercise 1D Ising model



average over every other spin.

$E = -J \sum_n \sigma_n \sigma_{n+1}$ $\quad E_{\square} = -J(\sigma_{2n}\sigma_{2n+1} + \sigma_{2n+1}\sigma_{2n+2})$

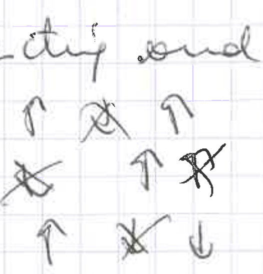
$\sum_{\{\sigma_{2n+1}\}} e^{-\beta E(\{\sigma_n\})} \Rightarrow e^{-\beta E'}$ ← it looks like 2n and 2n+2 now interact

sum over odd spins.

$E' = -J \sum_n \sigma_{2n}\sigma_{2n+2}$ But 2n+2 does not interact with 2n-2, so spin still interact with NN, so I still have a Boltzmann distribution. But with a different energy.

The absolute value of J' is always smaller than J. So as you zoom out the spins become less and less interacting and then the spins never order.

In 2D



things get complicated because even if it's NN then you have interaction along the diagonal.

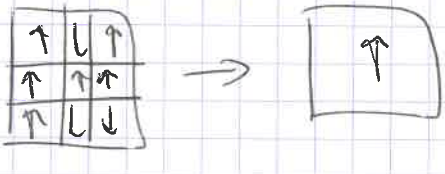
In 2D you get more terms and probably all these terms go away.

At the end for 2A terms only 2 survive.

Coarse grain:

- decimation (eg. every other spin)
- block spins

e.g.



- Fourier transform

$$\sigma(x_i) = \sum_{k} e^{ik\vec{x_i}} \sigma(\vec{k})$$

real space methods via numerical simulation you would do this

I ~~am~~ going to produce coarse grained variables

$$\tilde{\sigma}_\ell = z_\ell \sum_{|k|<\frac{1}{\ell}} e^{ik\vec{x_\ell}} \sigma(\vec{k})$$

momentum shell methods

pen & paper you would do this

The goal is not to make these constructions but to follow the probability distribution of the variables that are left.

Armed with these techniques go work on the methods you care about.

What is the problem?

real space $\xleftrightarrow{\text{needs}}$ neighbourhood

I need distance, what is a distance?

Real distance, probably not all the neurons in a given neighborhood matter?

If I want to do momentum shell? What is momentum

momentum shell ⟷ momentum?

We need to have candidates for what neighborhood of momentum mean?

These may not be the best solutions, they are just candidates.

version of this already exist in disordered system but problems here are severe.

I do know how to calculate the covariance matrix

$$\langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle \langle \sigma_j \rangle = C_{ij}$$

If your variables things live on a lattice and a ~~generally~~ positive

$$\boxed{\text{nearest neighbour of } j \longleftrightarrow \underset{j \neq i}{\text{argmax}}\, C_{ij}/\sqrt{C_{ii}C_{jj}}}$$

Pearson correlation coefficient

How would I change my length scale of averaging by a factor of 2:

find max C_{ij} $(i \neq j)$ $\quad (i, j^*(i))$ "NN" in this sense of i

$$\sigma_i + \sigma_{j^*(i)} = \tilde{\sigma}_i$$

repeat ... $\{\sigma_i\}_{i=1\dots N} \longrightarrow \{\tilde{\sigma}_i^{(1)}\}_{i=1\dots N/2}$

This statement makes sense for ~~all~~ one spin. Now you need to do this for all ~~of~~ them (pair them up). Here proposing to do this in a greedy way.

This is a coarse group strategy.

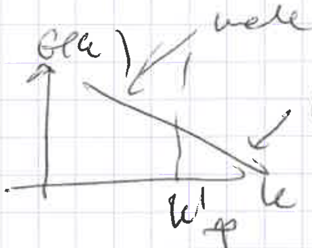
It depends on what my ~~[illegible]~~ observables are.

For momentum shell you want translational invariance

$$C_{ij}\, u_j^{\mu} = \lambda_\mu\, \underline{\underline{u_i^{\mu}}} \longleftarrow e^{i\vec{k}\cdot\vec{x}_i}$$

$$\lambda_\mu \Longleftrightarrow \int d^3r\; C(\vec{r})\, e^{i\vec{k}\vec{r}}$$

$$G(\vec{k})$$



this sounds like PCA

The question in PCA is where is the cut-off.

In most system the ~~[illegible]~~ spectrum is continuous.
So RG tells you you should ask what changes when you change the cut-off.

The proposal:

what corresponds to neighbourhood is the most correlated component

What corresponds to momentum are the principal components.

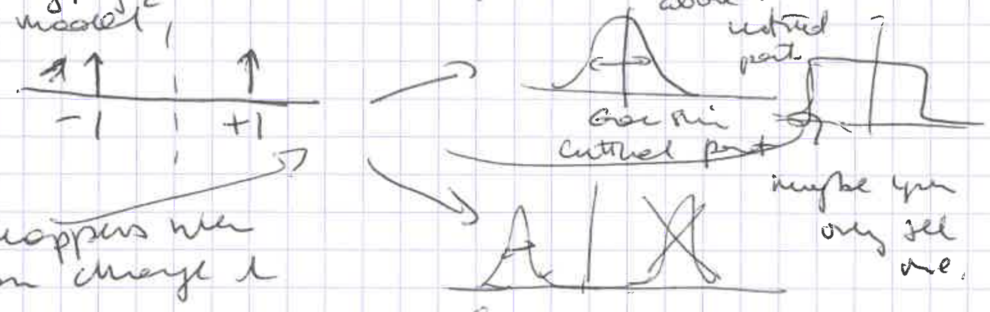
(caveat: all this is for one time point)

implement → how do you follow flow?

"flow" $(g_1, g_2, \ldots) \to$?

what is it for the Ising model,

$P_\ell(\tilde{\sigma}_1^\ell)$



People who have been doing Monte Carlo simulation have thought about this.

$$\langle \sigma_i^{(\ell)}(t)\, \sigma_i^{(\ell)}(t+\tau)\rangle_c = C_\ell(\tau) = C\left(\tau/\tau_c(\ell)\right)$$

If things are becoming simpler when you vary l.

dynamic scaling

$\tau_c \sim \ell^z$

These are the examples of the kinds of things you can look for.

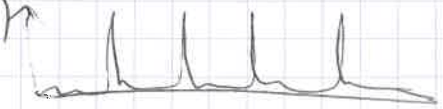
We are going to follow these coarse graining procedures

We are going to do it for real data for a mouse, fluorescent protein that lights up when calcium is high (action potential).

In most regions of the mouse brain nothing happens

unless the mouse behaves (let the mouse run on a styrophore ball),

→ virtual reality for a mouse.



or $\sigma_i = \{0, 1\}$ discrete variable.

Real space version

compute correlation matrix, list most correlated pair, combine.

normalize so that mean nonzero signal = 1

continue until N cells became N/2

iterate

Momentum Flow!

non spiking silence. probability that the coarse grained cluster size distribution is zero,

$$P_k(x) = \overbrace{P_0(k)\,\delta(x)} + [\underbrace{1-P_0(k)}]\,Q_k(x)$$

$$\int dx\, Q_k(x) = 1$$

probability that the coarse grained distribution is not zero.

experimentally you see

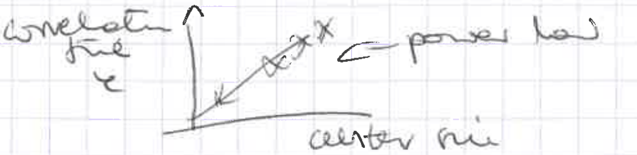
$$P(k) = \exp(-\alpha k^{\beta})$$

$$\beta \simeq 0.87 \underbrace{\pm 0.014}_{\text{within one mouse}} \underbrace{\pm 0.015}_{\text{within different mice}}$$

If you look at the continuous part of the distribution (the non-silent one).

The distribution of the coarse grained distribution is approaching a fixed distribution.

Dynamic scaling: if you rescale by the correlation time you see a collapse.



$\tau_c \sim k^{\tilde{z}}$

$\tilde{z} \sim$ [illegible]

now momentum space

→ distribution of coarse grained variables approaches a fixed non-Gaussian form.

and you can get dynamic scaling ~~that~~ $\tau_c \sim \xi^z$
and the exponents agree.