Estimating energy-based models with Score Matching

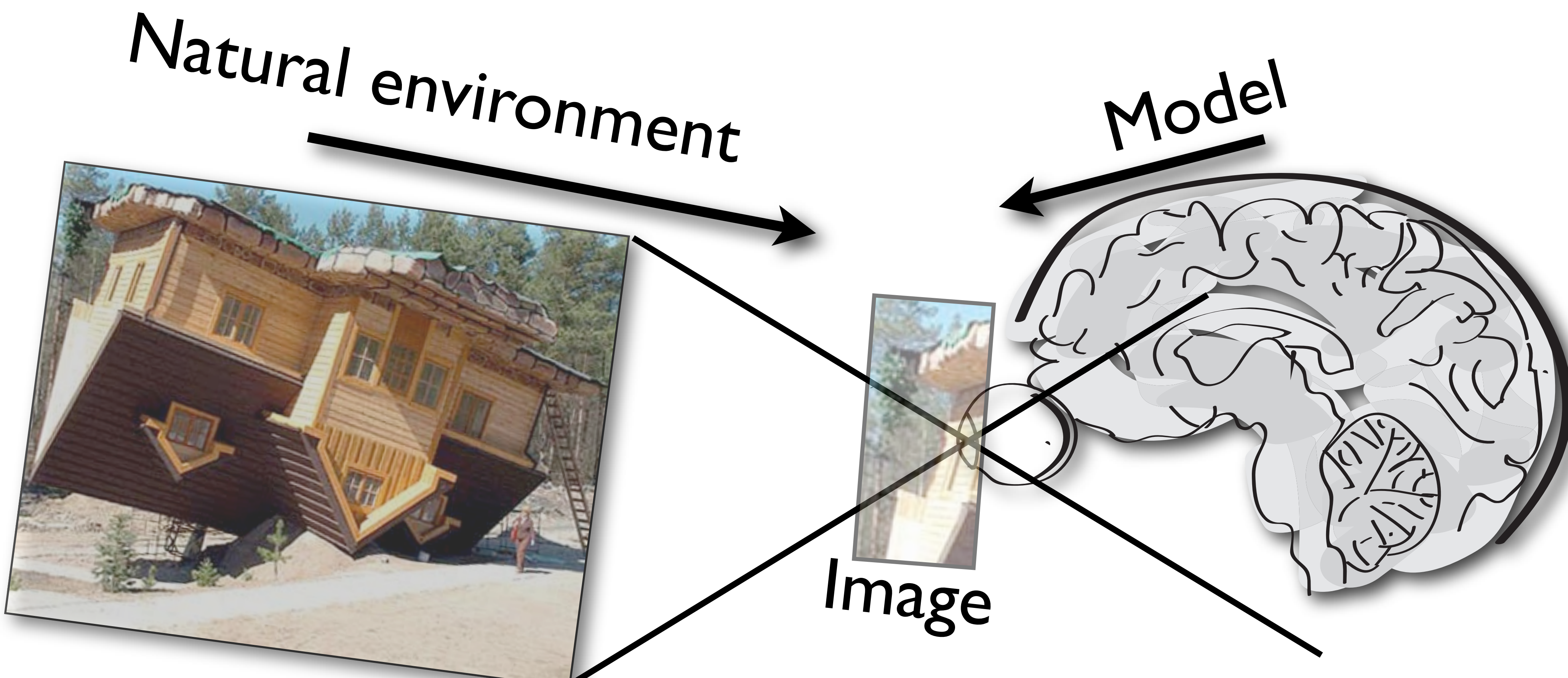
Urs Köster
UC Berkeley

Overview

- Unsupervised learning going beyond Maximum Likelihood
- Score Matching estimation
- Example: Overcomplete Product of Experts model

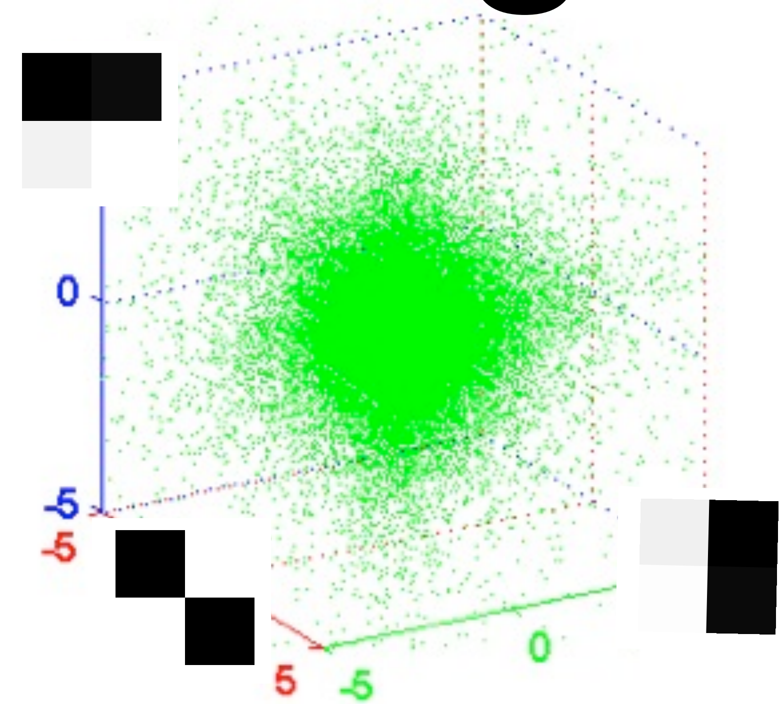
Unsupervised Learning

- The brain makes inferences about the world
- Requires a model of the environment
- How is the brain doing this?

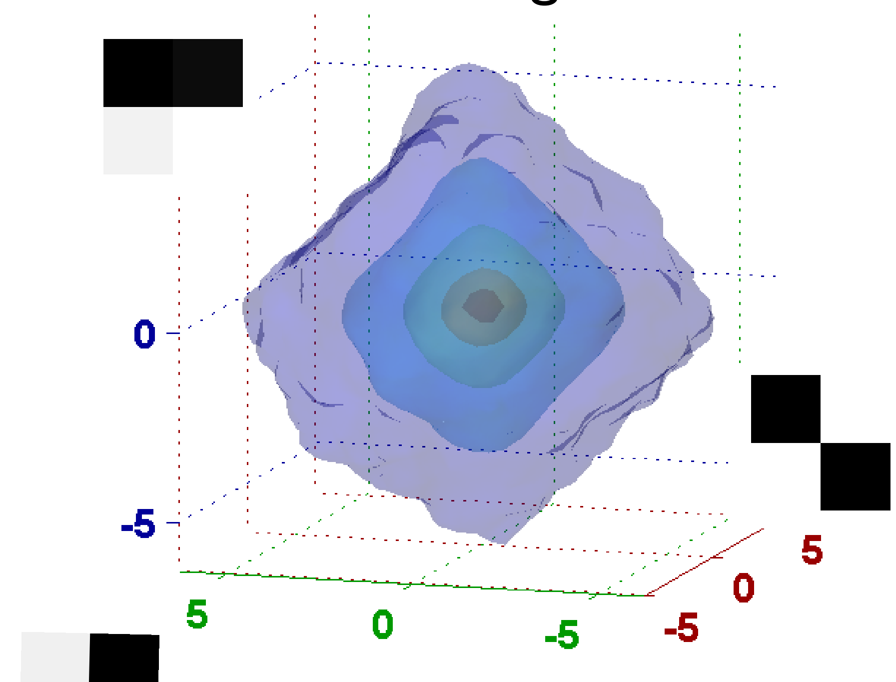


Unsupervised Learning

- Fit a function to observed data
- Assume the data is samples from a pdf $p_x(\xi)$ that cannot be observed directly
- Have a parametrized function $p_m(\xi,\theta)$ that we try to fit to the data samples x



Natural image data



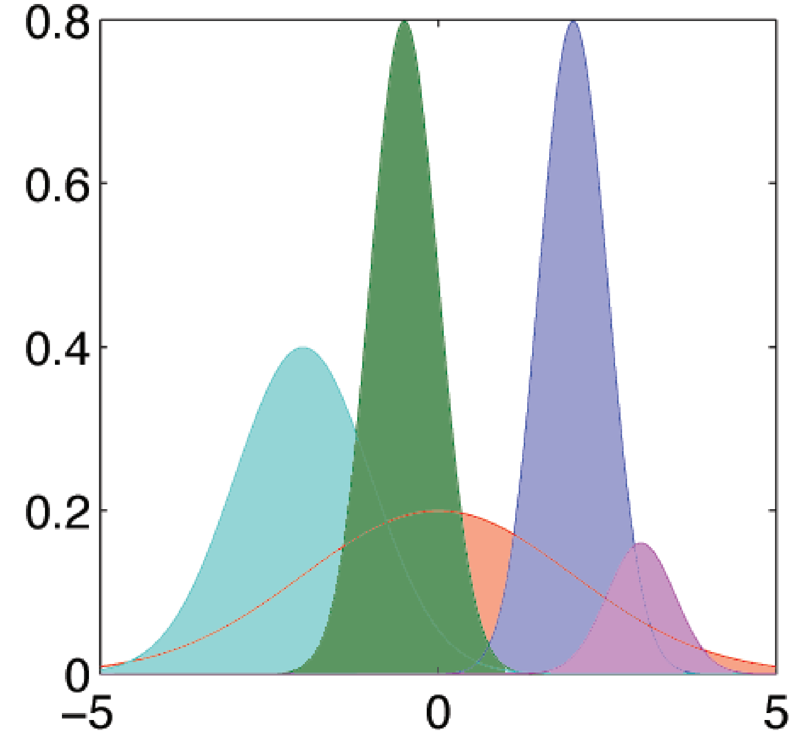
The “true” pdf $p_x(\xi)$

Maximum Likelihood Estimation

- Maximize the expected likelihood of the parameters given the observed data

$$\arg\max E\left[p_m(x|\theta)\right]$$

- for convenience, use $log(p_m)$ which give the same result since log is monotonic
- Works under the condition that the function p_m is constrained to have constant volume, i.e. is a pdf that integrates to unity.



volume of the pdf normalized to unity

- Limited family of models (ICA, L_p-spherical densities, no overcompleteness)

Energy based models

- What if the functional form of $log\ p_m$ does not permit normalization?
- This requires integrating over the whole space which is intractable in most cases.
- Model of the form
$$p_m(\xi) = 1/Z(\theta)\exp(-E(\xi,\theta))$$
- where $E(\xi, \theta)$ is the energy of the model and $Z(\theta)$ is the partition function required so the pdf integrates to unity
- We cannot fit $p_m(\xi)$ to observations from $p_x(\xi)$ since we only know the model pdf up to a constant!

Score Matching

Score Matching

- Circumvent the problem by fitting the gradients of the model to the gradient of the data pdf.
- Model score function

$$\psi_i(\xi, \theta) = \frac{d}{d\xi_i} \log p_m(\xi|\theta)$$

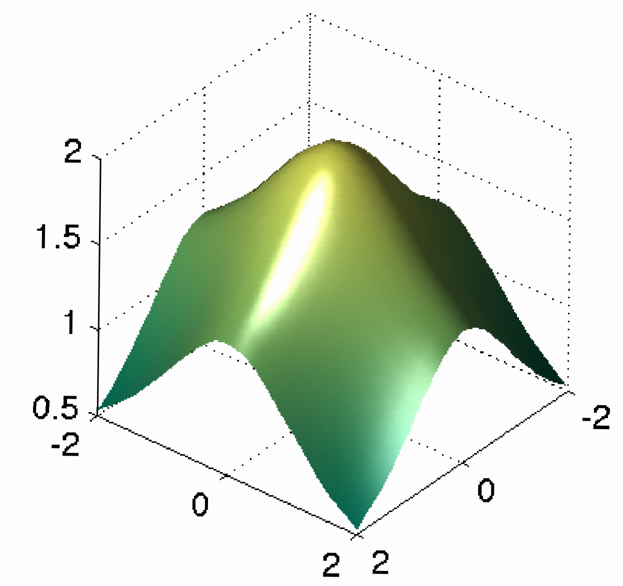
- and data score function

$$\psi_{x,i}(\xi) = \frac{d}{d\xi_i} \log p_x(\xi)$$

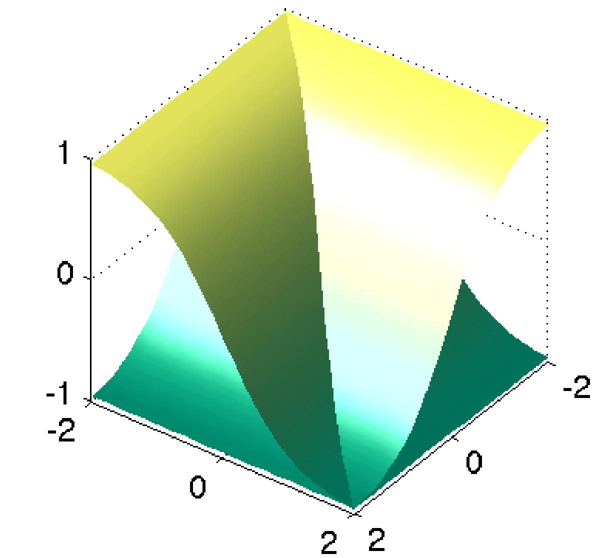
- Minimize the expected distance between the two (match the score functions)

$$J = \frac{1}{2} \int p_x(\xi) ||\psi(\xi, \theta) - \psi_x(\xi)||^2 d\xi$$

- which gives a consistent estimator if the pdf is smooth and the data follows the model.



2D Student-t log pdf



Components of the Score function

Score Matching

- Hold on: How does this help if we don't know the data score function $\psi_x(\xi)$?
- Massage the "Score Match" into a format that does not depend on this intractable quantity

The maths

- Start by expanding the squared distance

$$||\psi(\xi,\theta)-\psi_x(\xi)||^2 = \psi_x(\xi)^T\psi_x(\xi) + \sum_i \psi_i(\xi,\theta)^2 - 2\psi_i(\xi,\theta)\psi_{x,i}(\xi)$$

- which gives us three terms under the expectation.
- The first one can be ignored since it's constant that does not depend on the parameters
- The second term is the expectation of the squared model score function (easy to compute)
- The third term however has nasty dependency on the data score function

More maths

- Working on
$$-2\sum_i \int p_x(\xi)\psi_i(\xi,\theta)\psi_{x,i}(\xi)d\xi$$
- first use the definition of the score function,
$$\psi_{x,i}(\xi) = \frac{d}{d\xi_i}\log p_x(\xi) = 1/p(\xi)\frac{d}{d\xi_i}p_x(\xi)$$
- So we get
$$-2\sum_i \int p_x(\xi)\psi_i(\xi,\theta)\frac{1}{p_x(\xi)}\frac{d}{d\xi_i}p_x(\xi)d\xi$$
- and cancel $p(\xi)$. Then use integration by parts to switch the differentiation operator to the score function
$$2\sum_i \int \frac{d}{d\xi_i}\psi_i(\xi,\theta)p_x(\xi)d\xi$$
- flipping the sign in the process. We are left with an expectation containing only the model score function
- The constant of integration goes to zero as p does

Assembling the pieces

- The objective function is now

$$J = \frac{1}{2}\sum_i \int p(\xi)\psi_i(\xi)^2 d\xi + \sum_i \int \frac{d}{d\xi_i}\psi_i(\xi,\theta)p(\xi)d\xi + C$$

- If we replace integrals by sample expectations

$$J = \frac{1}{2}\sum_{t=1}^{T}\sum_i \psi_i(x(t))^2 + \frac{d}{dx_i}\psi_i(x(t),\theta) + C$$

- This is the final objective function

- Intuitively, first term acts like unnormalized likelihood, second term like normalization

- Gives a consistent estimator in terms of simple expectations of non-normalized model pdf

Caveats

- Taking the second derivative requires reasonably smooth nonlinearities
- Taking the second derivative also requires reasonably simple energy functions

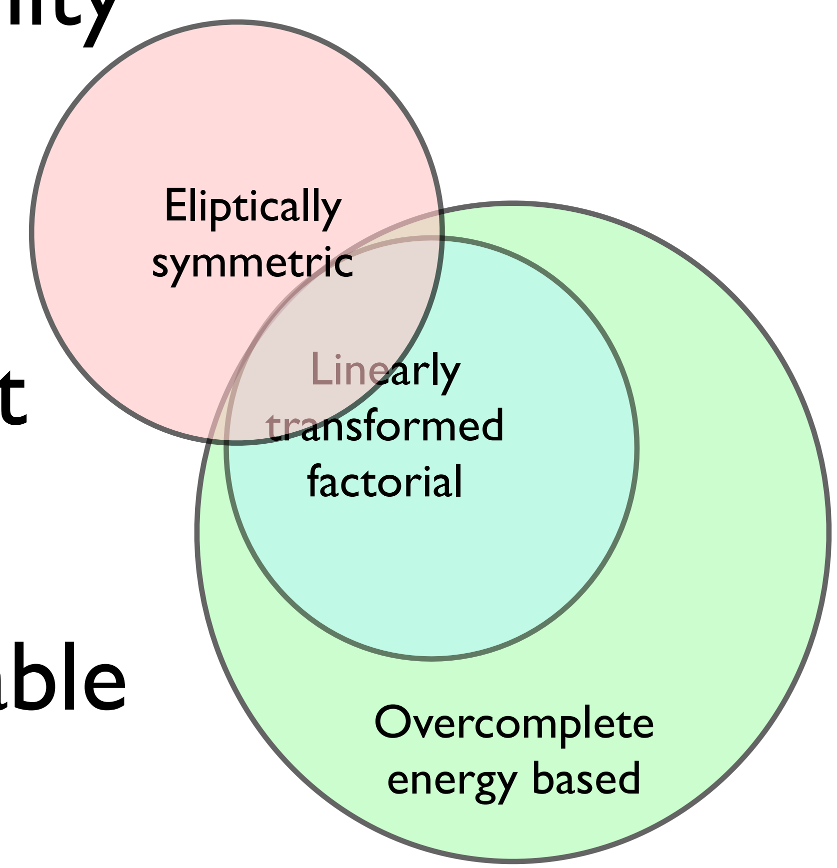
Overcomplete Products of Experts

PoE model

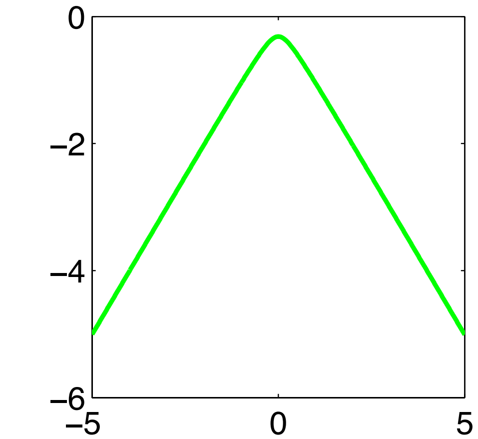
- Prototypical energy-based model

$$E = \sum_{k}^{K>n} \alpha_k g(\mathbf{w}_k^T \mathbf{x})$$

- Intuition: Slice up probability space by defining regions of low probability (high energy).
- Generalization of factorial models that does not assume independent sources
- Normalization constant is intractable



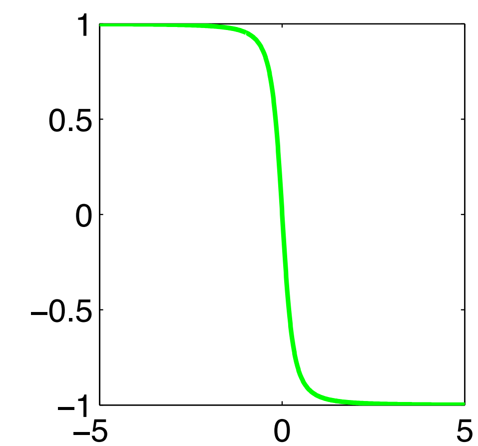
PoE model



Logistic energy function

- Energy is defined as the sum of a number of weighted potential energy functions

$$E = \sum_k \alpha_k g(\mathbf{w}_k^T \mathbf{x})$$



First derivative

- g can correspond to something like a Cauchy or Logistic distribution (heavy tails)

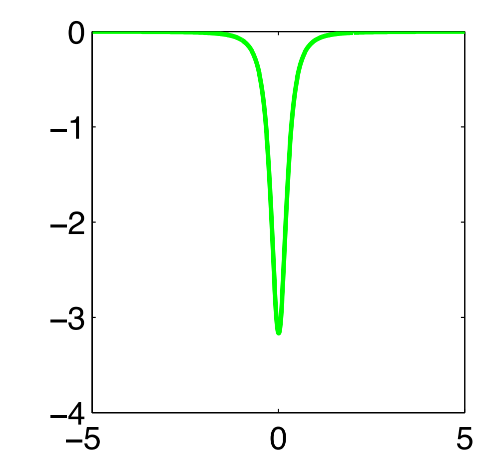
$$g(u) = \log\cosh(u) \qquad g'(u) = \tanh(u) \qquad 1 - \tanh(u)^2$$

- the score function corresponding to this model is

$$\psi_i = -\frac{d}{dx_i} E = -\sum_k \alpha_k g'(\mathbf{w}_k^T \mathbf{x}) w_{ki}$$

- and the second derivative of the energy

$$\psi_i' = -\frac{d^2}{dx_i^2} E = -\sum_k \alpha_k g"(\mathbf{w}_k^T \mathbf{x}) w_{ki}^2$$



Second derivative

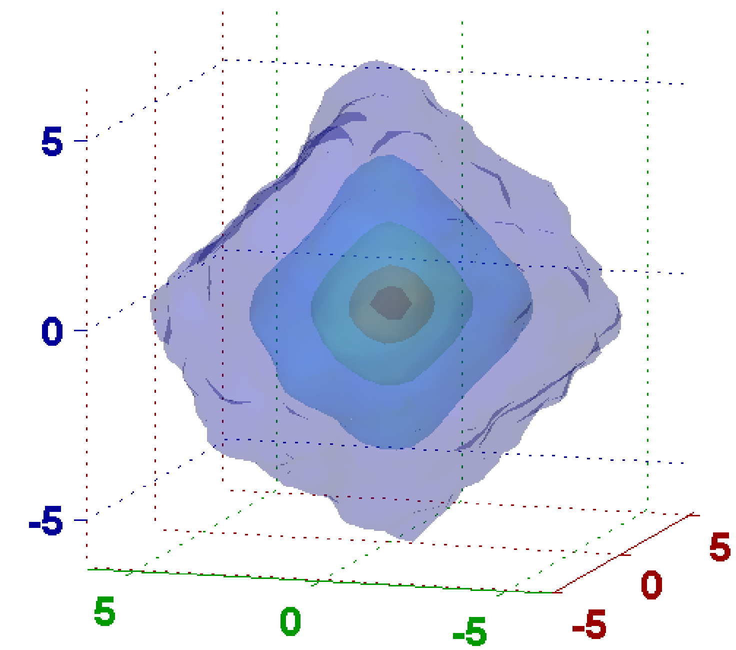
PoE model

- Putting this into the objective
- gives

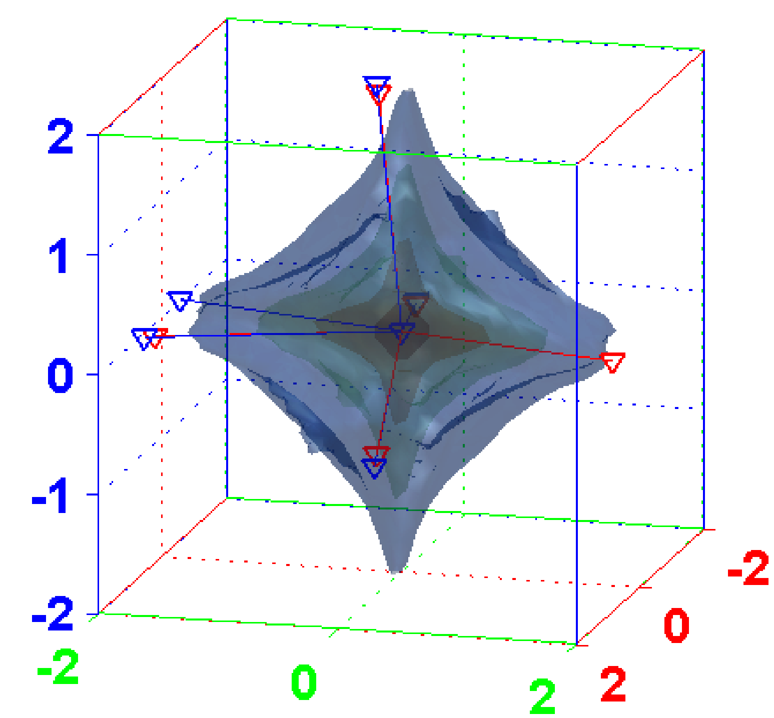
$$J = \sum_{t=1}^{T} \sum_{i} \frac{1}{2} \psi_i(x)^2 + \frac{d}{dx_i} \psi_i(x, \theta)$$

$$J = \sum_{t=1}^{T} \sum_{i} \frac{1}{2} \left(\sum_{k} \alpha_k g'(\mathbf{w}_k^T \mathbf{x}(t)) w_{ki} \right)^2 + \sum_{k} \alpha_k g"(\mathbf{w}_k^T \mathbf{x}(t)) w_{ki}^2$$

- to estimate the filters and α's (expert parameters) compute the gradients and plug in to any optimization package.
- Example with 10 filters for 3D image data: Capture something *interesting* about the structure



Natural image data



Model fit

Summary

- When would you use Score Matching?
- Alternatives such as Contrastive Divergence, Noise Contrastive Estimation and Minimum Probability Flow (tomorrow!) exist. Which is best?
- SM is not based on sampling, so none of the problems with Monte Carlo methods
- Depending on the model (hierarchical, MRF ...) computing gradients can be slightly to very tedious
- Gradient estimation can use efficient methods like l-BFGS
- Objective function "Score Match" allows model comparison without having to calculate likelihoods