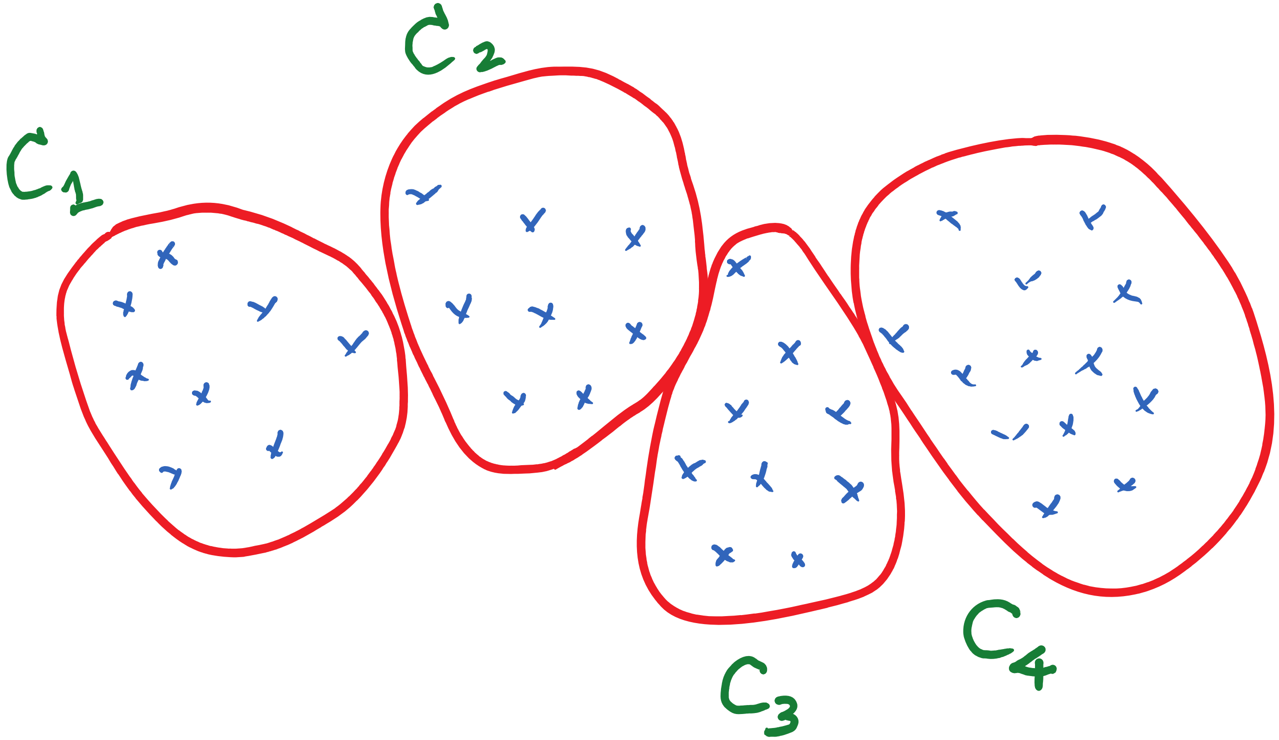
Applications of Information Geometry

Clustering based on Divergence: Cluster Center
Stochastic Reasoning: Belief Propagation in Graphical Model
Support Vector Machine
Bayesian Framework and Restricted Boltzmann Machine
Natural Gradient in Multilayer Perceptron Learning
Independent Component Analysis
Sparse Signal Analysis and Minkovskian Gradient
Convex Optimization

Clustering of Points

$$D = \{\theta_1, \theta_2, \theta_3, \ldots, \theta_N\}$$

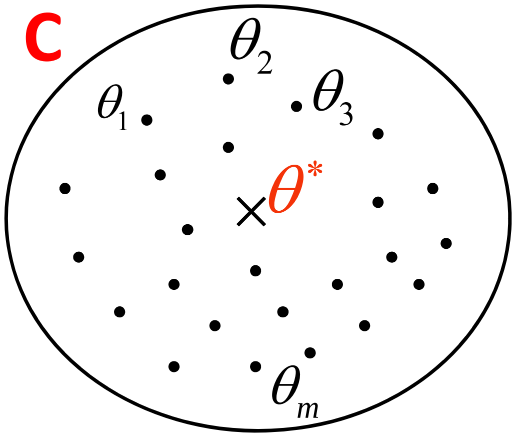


Clustering : center of a cluster of points

$$C = \{\theta_1, \cdots, \theta_m\}$$

Center of C

$$\theta^* = \arg\min_{\theta} \sum_i D[\theta, \theta_i]$$



η^* is simply the arithmetic average in the dual coordinate system

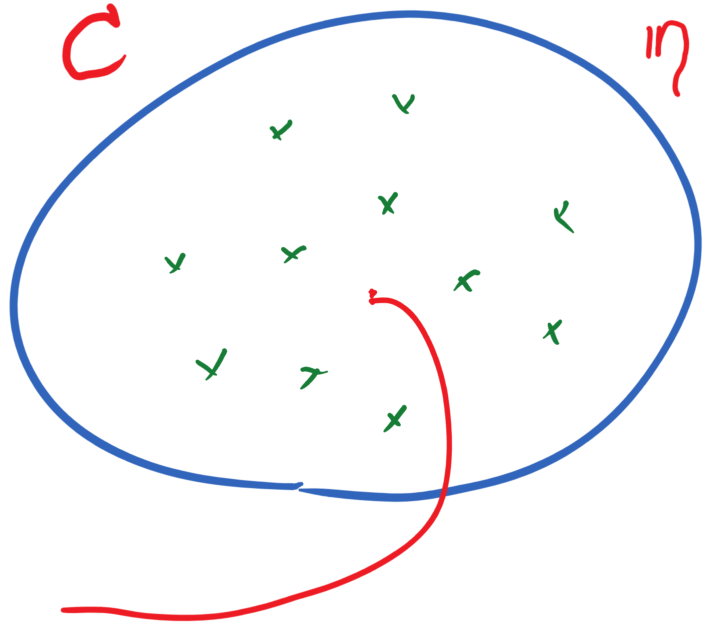
$$D[\theta:\theta_i] = \psi(\theta) + \varphi(\eta_i) - \theta \cdot \eta_i$$

$$\partial_\theta D[\theta:\theta_i] = \eta - \eta_i$$

$$\sum (\eta - \eta_i) = 0$$

$$\eta^* = \frac{1}{m} \sum \eta_i$$

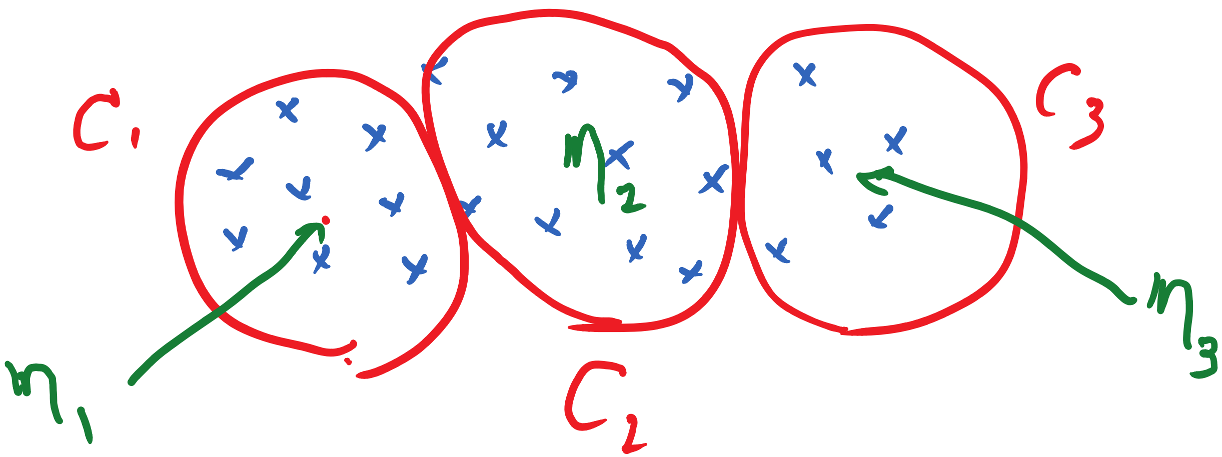
k-means: clustering algorithm



$$\eta^* = \frac{1}{m} \sum \eta_i$$

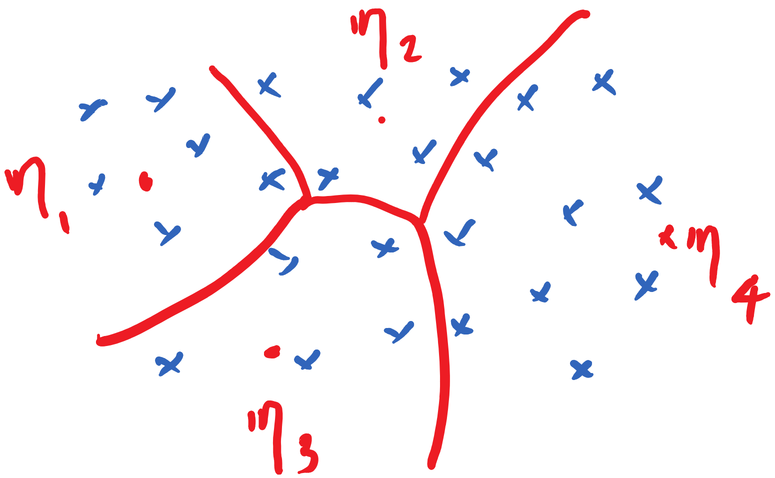
k-means: clustering algorithm

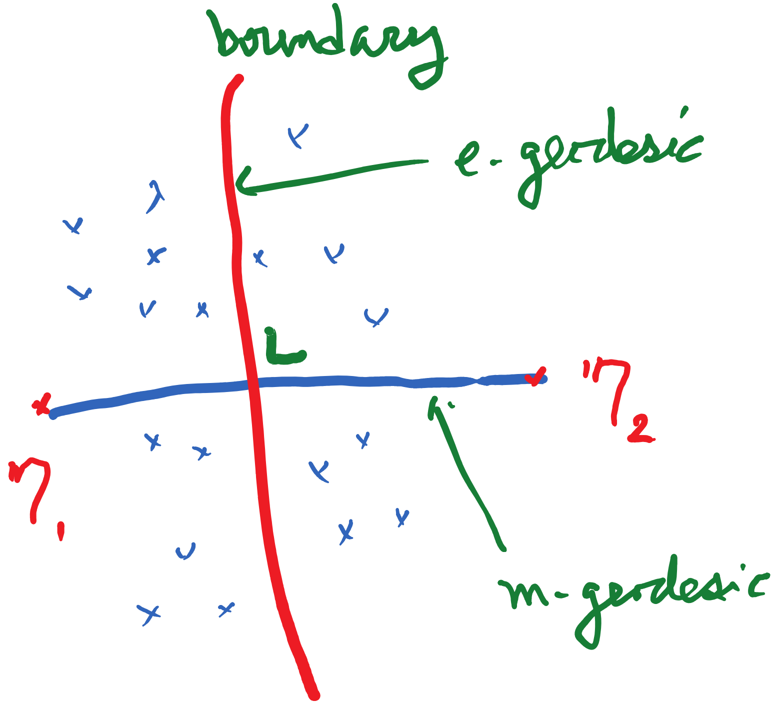
1. Choose k cluster centers
2. Classify patterns of D into clusters
3. Calculate new cluster centers



Voronoi Diagram: Boundaries of Clusters

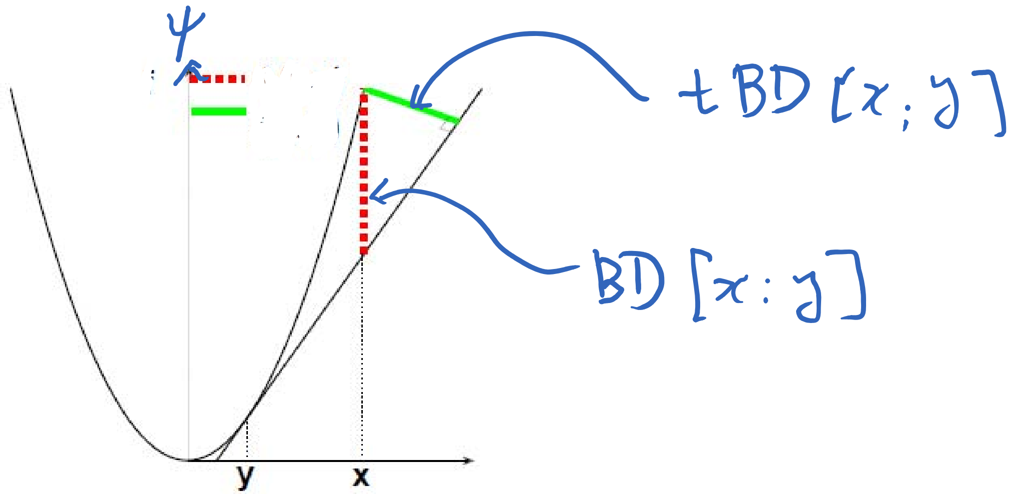
$$D[\eta : \eta_1^*] = D[\eta : \eta_2^*]$$





Total Bregman divergence (Vemuri)

$$\mathrm{tBD}(p:q) = \frac{\psi(p)+\varphi(q)-\theta_p\cdot\eta_q}{\sqrt{1+|\nabla\psi(q)|^2}}$$



t-center of C η^*

$$\eta^* = \frac{\sum w_i \eta_i}{\sum w_i}$$

$$w_i = \frac{1}{\sqrt{1 + \left\| \nabla \psi(\eta_i) \right\|^2}}$$

Conformal change of divergence

$$\tilde{D}(p:q) = \sigma(q) D[p:q]$$

$$\tilde{g}_{ij} = \sigma(p) g_{ij}$$

$$\tilde{T}_{ijk} = \sigma(T_{ijk} + s_k g_{ij} + s_j g_{ik} + s_i g_{jk})$$

$$s_i = \partial_i \log \sigma$$

t-center is robust

$$E^* = \{\eta_1, \cdots, \eta_n; \boldsymbol{y}\}$$

$$\tilde{\eta}^* = \eta^* + \varepsilon \boldsymbol{z}(\eta^*; \boldsymbol{y}), \quad \varepsilon = \frac{1}{n}$$

influence function $\boldsymbol{z}(\eta^*; \boldsymbol{y})$

$|\boldsymbol{z}| < c$ as $|\boldsymbol{y}| \to \infty$: robust

$$\text{minimize } \frac{1}{N+1}\left(\sum \frac{D[\eta:\eta_i]}{w_i} + \frac{D[\eta:\mathrm{y}]}{w_{N+1}}\right)$$

$$\boldsymbol{z}\left(\eta^*, \boldsymbol{y}\right) = G^{-1} \frac{\nabla\psi\left(\theta_y\right) - \nabla\psi\left(\eta^*\right)}{w\left(\boldsymbol{y}\right)}$$

$$\boldsymbol{z}\left(\eta^*, \boldsymbol{y}\right) = G^{-1} \frac{\boldsymbol{y}}{\sqrt{1+\left\|\boldsymbol{y}\right\|^2}}$$

Robust: $\boldsymbol{z}$ is bounded

$$\boldsymbol{z}\left(\eta^*, \boldsymbol{y}\right) = \boldsymbol{y}$$

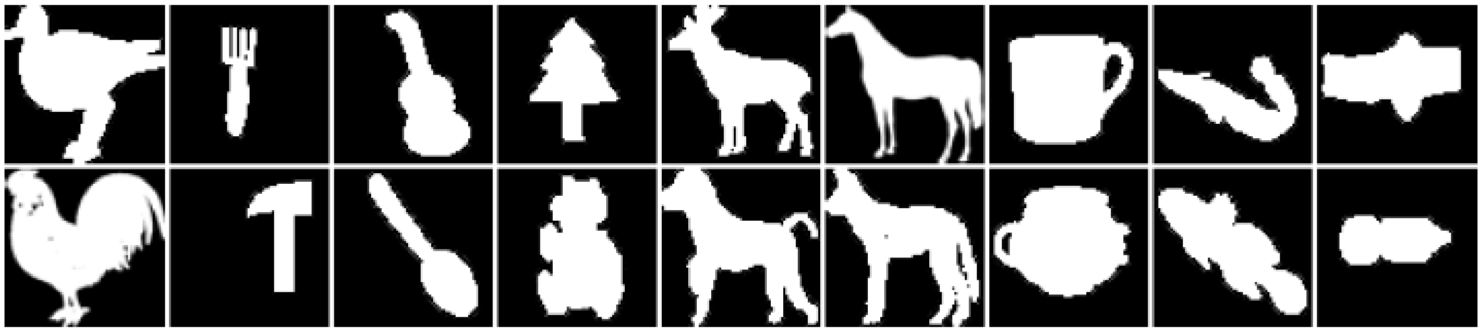
$$\frac{\nabla\psi\left(\theta_y\right)}{w\left(\boldsymbol{y}\right)} = \frac{\nabla\psi\left(\theta_y\right)}{\sqrt{1+\left\|\nabla\psi\left(\theta_y\right)\right\|^2}} < \infty$$

Euclidean case $\quad f = \frac{1}{2}\left|\boldsymbol{x}\right|^2$

$$w\left(\boldsymbol{y}\right) > 1$$

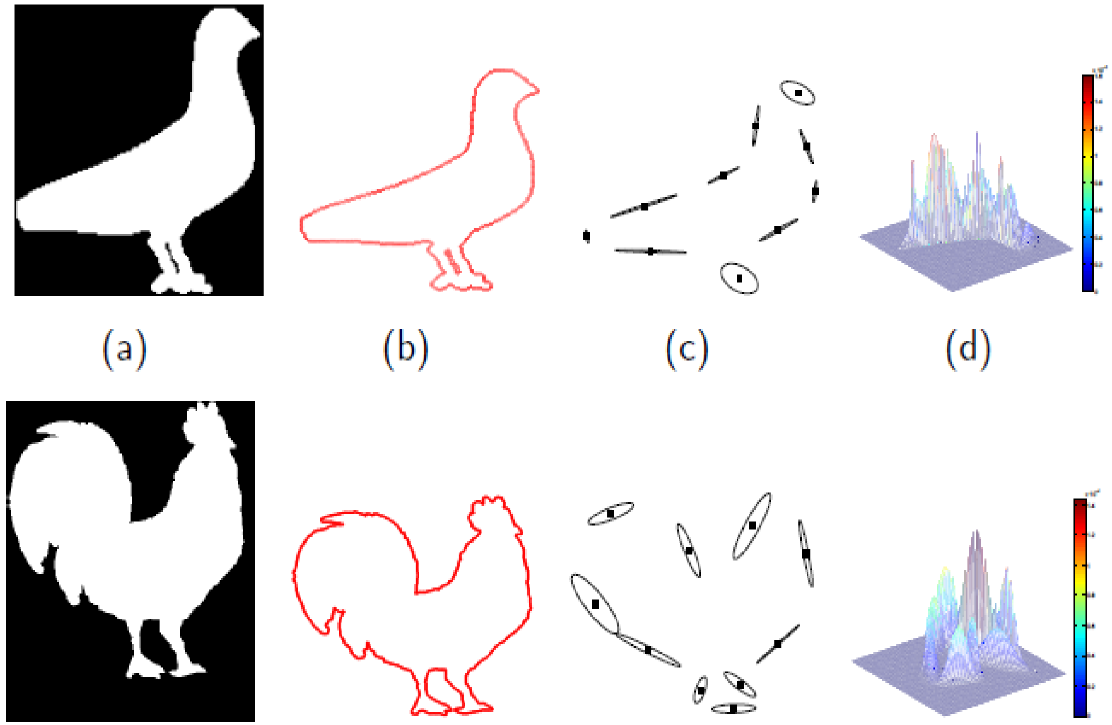
MPEG7 database

- **Great intraclass variability, and small interclass dissimilarity**.



Shape representation

A shape is represented using a mixture of Gaussians from the aligned boundary points.



(a) (b) (c) (d)

(e) (f) (g) (h)

First clustering then retrieval

Retrieval in the whole MPEG-7 database

First clustering and then retrieval.

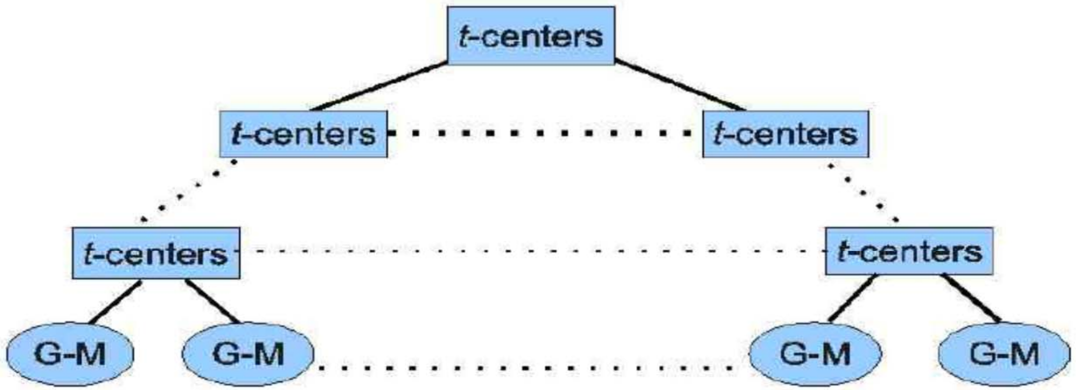


Figure: k-Tree diagram. G-M: mixture of Gaussians. Every key is a mixture of Gaussians. Each key in the inner nodes is the t-center of all keys in its children nodes. The key of a leaf is a mixture of Gaussians corresponding to an individual shape.

Other TBD applications

Diffusion tensor imaging (DTI) analysis [Vemuri]

- **Interpolation**
- **Segmentation**

Baba C. Vemuri, Meizhu Liu, Shun-ichi Amari and Frank Nielsen, *Total Bregman Divergence and its Applications to DTI Analysis*, IEEE TMI, to appear

Information Geometry of Stochastic Reasing

Belief Propagation in Graphical Model

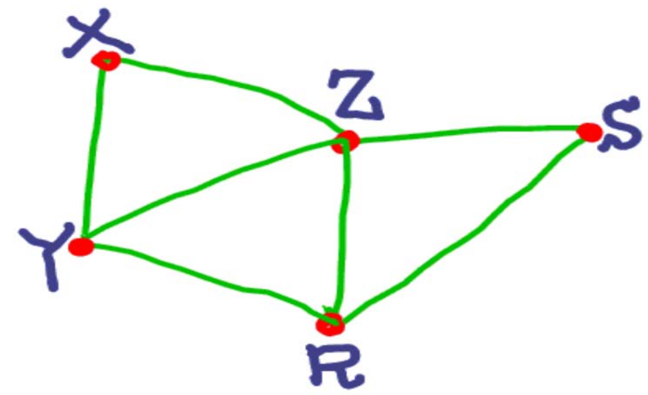
- Shun-ichi Amari (RIKEN BSI)
- Shiro Ikeda (Inst. Statist. Math.)
- Toshiyuki Tanaka (Kyoto U.)

Stochastic Reasoning: Graphical Model

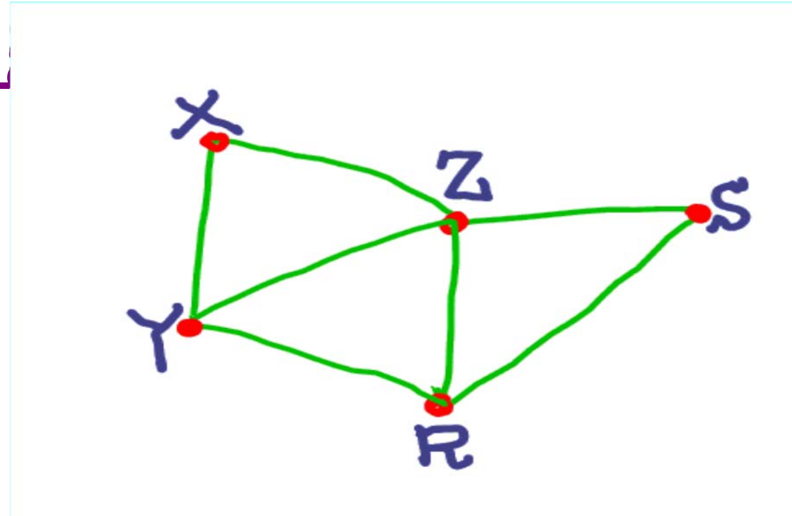
$$p(x, y, z, r, s)$$

$$p(x, y, z \mid r, s)$$

$$x, y, z, \ldots = 1, -1$$



Stochastic Reasonin



$q(x_1, x_2, x_3, \ldots \mid \text{observation})$

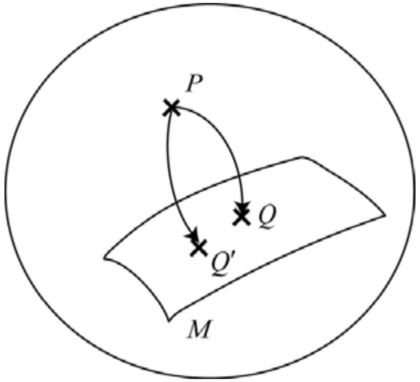
$\mathbf{X} = (x_1\ x_2\ x_3\ \ldots..)$ $\quad x = 1, -1$

$\mathbf{X} = \text{argmax}\ q(x_1, x_2, x_3, \ldots..)$ maximum likelihood

$X_i = \text{sgn}\ E[x_i]$ least bit error rate estimator

Mean Value

Marginalization:
projection to independent distributions

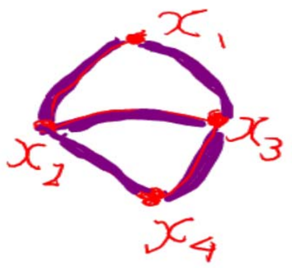


$$\Pi_0 q(\mathbf{x}) = q_1(x_1)q_2(x_2)\ldots q_n(x_n) = q_0(\mathbf{x})$$

$$q_i(x_i) = \int q(x_{1,} \ldots, x_n) dx_1 .. d\breve{x}_i .. dx_n$$

$$\eta = \mathbf{E}_q[\mathbf{x}] = \mathbf{E}_{q_0}[\mathbf{x}]$$

cliques



$$q(\boldsymbol{x}) = \exp\left\{ \sum k_i \cdot x_i + \sum_{r=1}^{L} c_r(\boldsymbol{x}) - \psi_q \right\}$$

$$c_r(\boldsymbol{x}) = c_r x_{i_1} \cdots x_{i_s}, \qquad r = (i_1 \cdots i_s)$$

$$x_i = \{1, -1\} \qquad r = (i_1, i_2)$$

Boltzmann machine, spin glass, neural networks
Turbo Codes, LDPC Codes

Computationally Difficult

$$q(\boldsymbol{x}) \rightarrow \eta = E[\boldsymbol{x}]$$

$$q(\boldsymbol{x}) = \exp\left\{\sum c_r(\boldsymbol{x}) - \psi_q\right\}$$

mean-field approximation

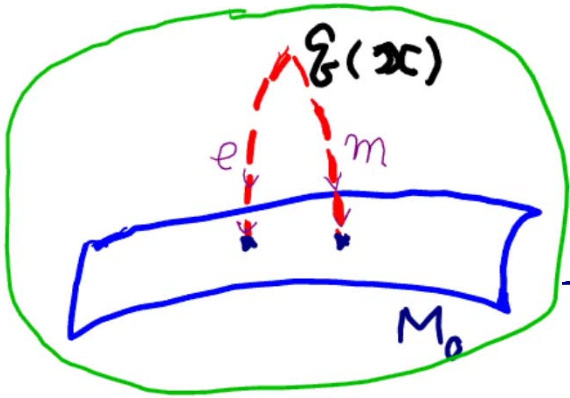
belief propagation

tree propagation, CCCP (convex-concave)

Information Geometry of Mean Field Approximation

- m-projection
- e-projection

$$D[q:p]=\sum_{x} q(x)\log\frac{q(x)}{p(x)}$$



$$M_0=\{\Pi_i p_i(x_i)\}$$

Π_0^m q = argmin D[q:p]

Π_0^e q = argmin D[p:q]

$p(x)\in M_0$

m-projection keeps the expectation of x

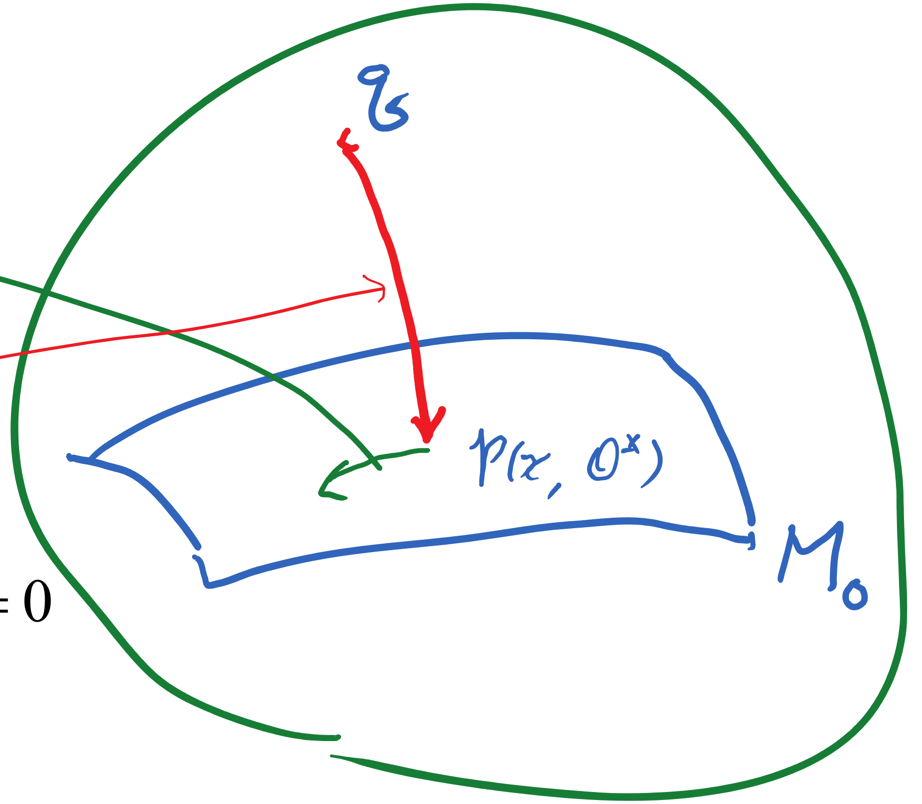
$$p(\mathrm{x}, \theta^*) = \prod q \in M_0$$

$$\frac{\partial}{\partial \theta} p(\mathrm{x}, \theta^*) = x - \eta^*$$

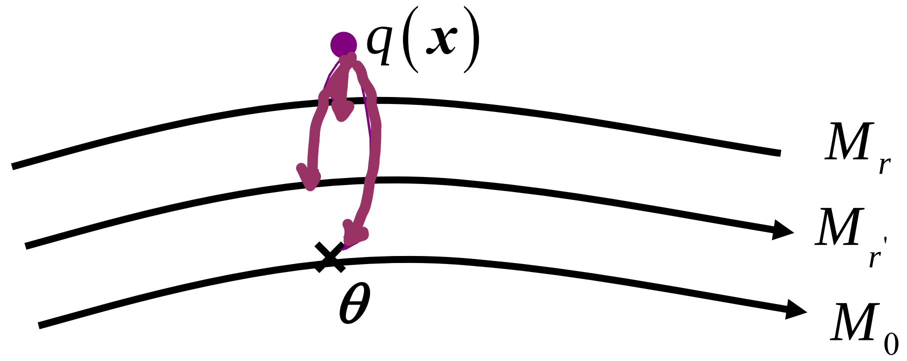
$$t(x) = \frac{q(x) - p^*(x)}{p^*(x)}$$

$$< t(x), x - \eta^* >_{p^*} = \sum_x (x - \eta^*) q(x) - p^*(x) = 0$$

$$E_q[x] = E_{p^*}[x]$$



Information Geometry



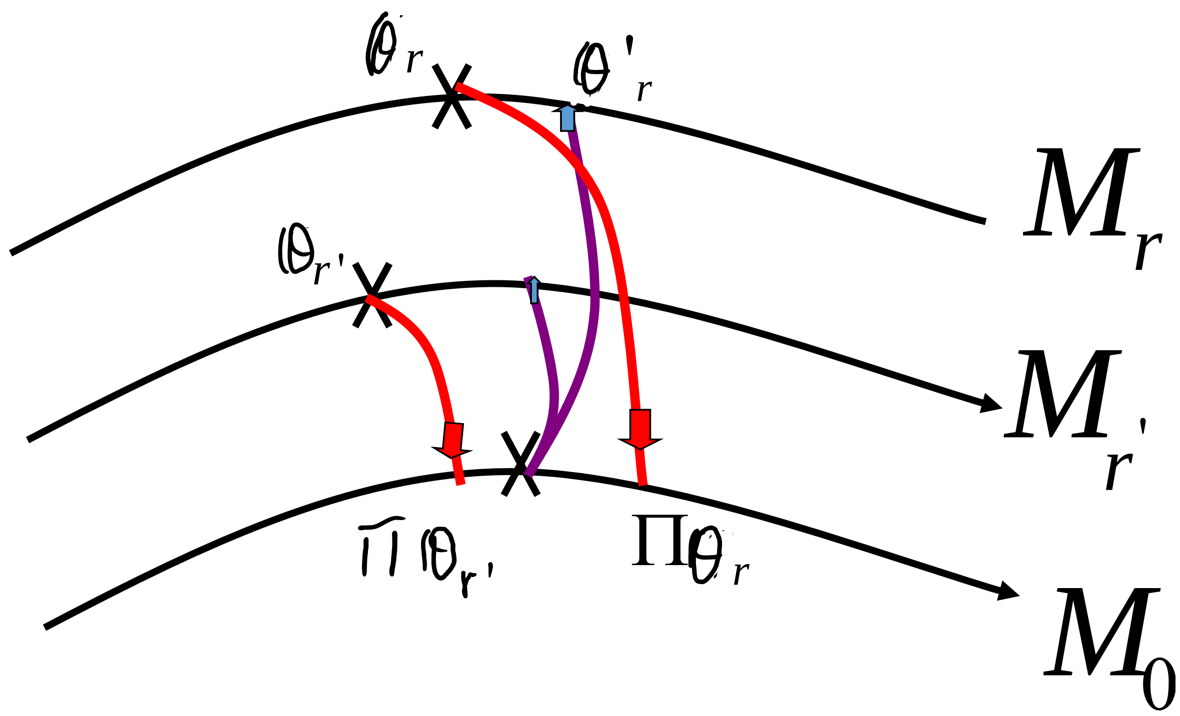
$$q(x) = \exp\left\{\sum c_r(x) - \phi\right\}$$

$$M_0 = \left\{p_0\left(\boldsymbol{x}, \boldsymbol{\theta}_0\right)\right\} = \exp\left\{\boldsymbol{\theta}_0 \cdot \boldsymbol{x} - \psi_0\right\}$$

$$M_r = \left\{p_r\left(\boldsymbol{x}, \theta_r\right) = \exp\left\{c_r\left(\boldsymbol{x}\right) + \theta_r \cdot \boldsymbol{x} - \psi_r\right\}\right\}$$

$$r = 1, \cdots, L$$

Belief Prop Algorithm



Belief Propagation

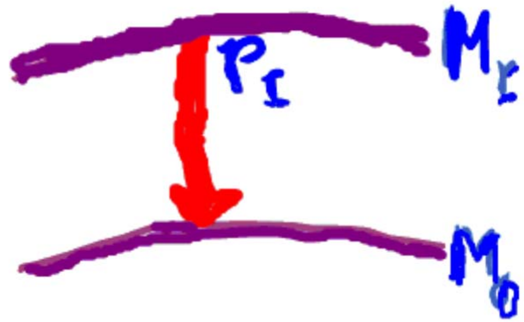
$$p(x,\xi_r) = \exp\{c_r(x) + \xi_r \cdot x - \psi_r\}$$

$$\Pi_0 p_r(x,\theta_r^t) \Rightarrow p_r(x,\tilde{\theta}_0^t)$$

$$\xi_r^{t+1} = \Pi_0 p_r(x,\theta_r^{\,t}) - \theta_r^{\,\mathrm{t}} \quad : \quad \text{belief for } c_r(x)$$

$$\theta_r^{t+1} = \sum_{r'\neq r} \xi_{r'}^{t+1}$$

$$\boldsymbol{\theta}_0^{t+1} = \sum \xi_r^{\,t+1}$$



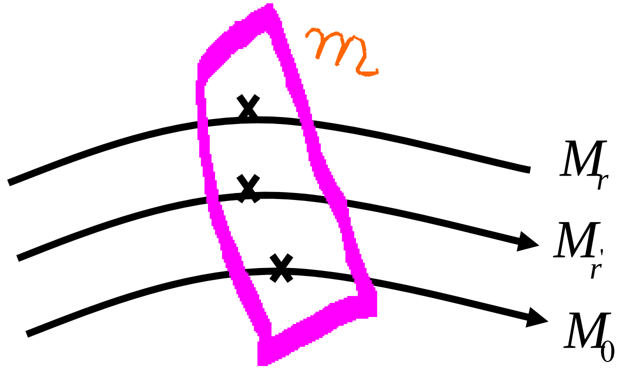
Equilibrium of BP $\left(\boldsymbol{\theta}^{*}, \boldsymbol{\xi}_r{}^{*}\right)$

1) *m*-condition

$$\boldsymbol{\theta}^{*} = \Pi_0 p_r\left(\boldsymbol{x}, \theta^{*}{}_r\right)$$

m-flat submanifold $M\left(\theta^{*}\right)$

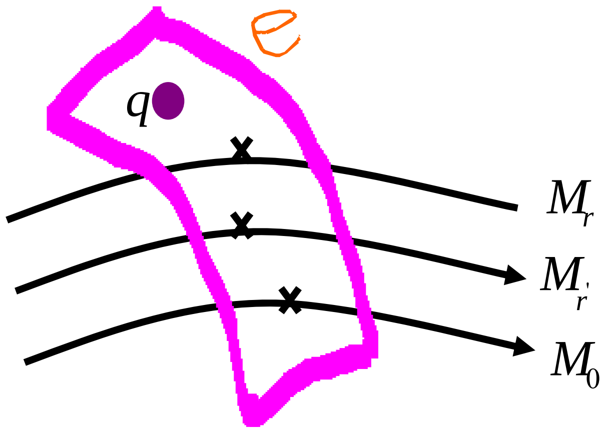
$\xi_1(\theta')$



2) *e*-condition

$$\boldsymbol{\theta}^{*} = \sum \boldsymbol{\xi}^{*}{}_r$$

$q\left(\boldsymbol{x}\right) \in e$-flat submanifold



Belief Propagation e-condition OK

$$(\boldsymbol{\theta};\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\ldots,\boldsymbol{\xi}_L), \qquad \boldsymbol{\theta}' = \sum \boldsymbol{\xi}'_r$$

$$(\theta_1,\theta_2,\ldots,\theta_L) \to (\theta'_1,\theta'_2,\ldots,\theta'_L)$$

CCCP **m-condition OK**

$$\boldsymbol{\theta}_0 \to \boldsymbol{\theta}_r : \prod p_0(x,\theta_r) = p_r(x,\theta_r)$$

$$\boldsymbol{\theta}_0' = \sum \xi_r$$

Convex-Concave Computational Procedure (CCCP) : A. Yuille

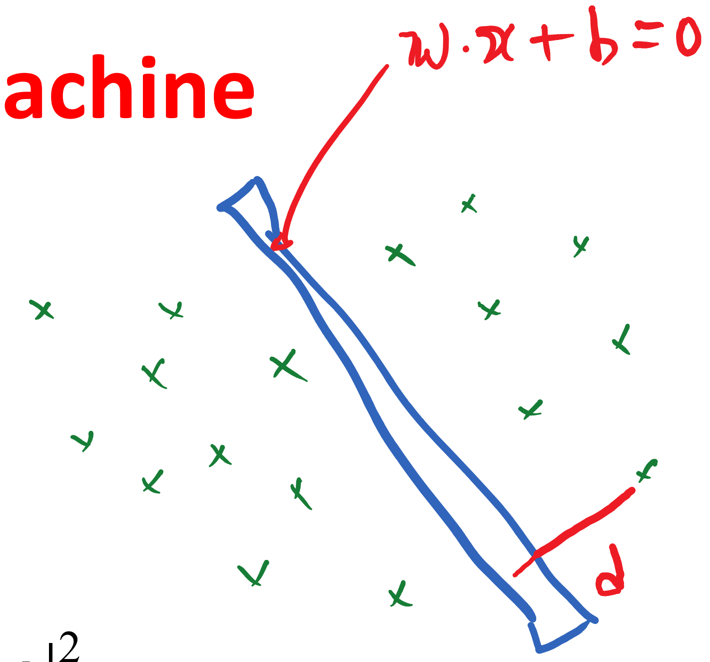
$$F(\theta) = F_1(\theta) - F_2(\theta)$$

$$\nabla F_1(\theta^{t+1}) = \nabla F_2(\theta^t)$$

Elimination of double loops

Geometry of Support Vector Machine

modification of kernel



Simple perceptron: decision surface

$$f(x) = w \cdot x + b$$

$$d = \frac{|w \cdot x + b|}{|w|}$$

minimize $|w|^2$

constraint $y_i(w \cdot x_i + b) \ge 1$

$$L(w,b,\alpha)=\frac{1}{2}|w|^2-\sum\alpha_i(w\cdot x_i+b)$$

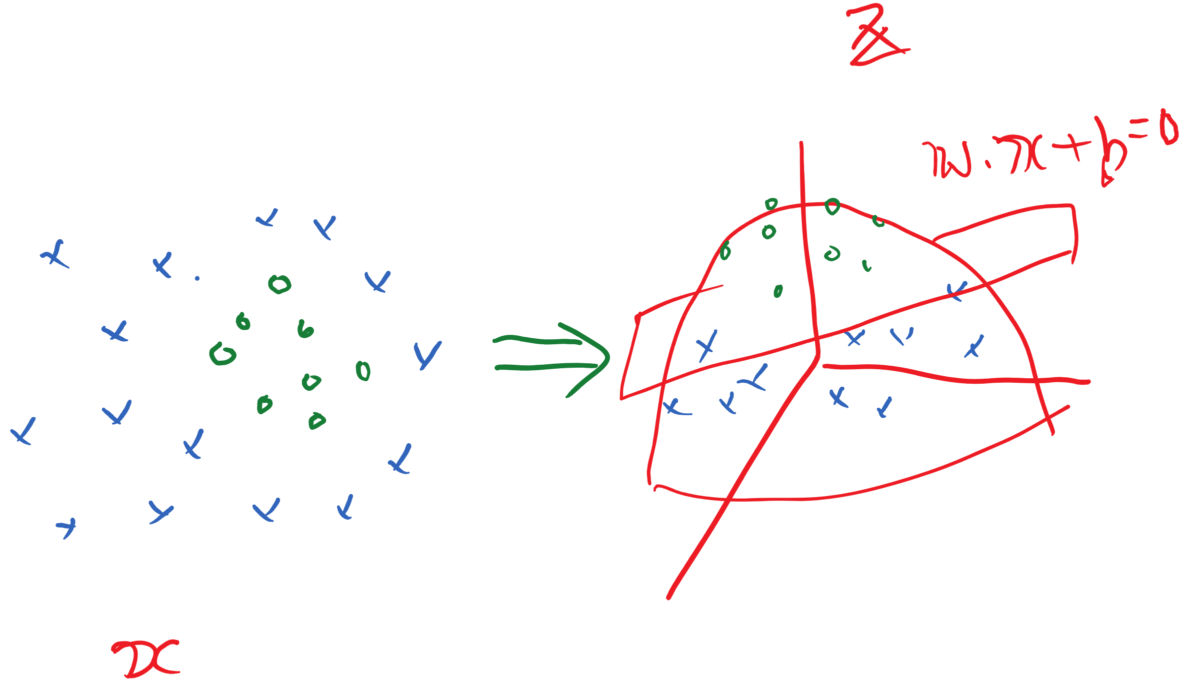
$$\sum\alpha_i y_i=0: \qquad \text{support vector } x_i:\ \alpha_i\neq 0$$

$$w=\sum\alpha_i y_i x_i \qquad f(x,w)=\sum\alpha_i y_i x_i+b$$

Embedding in higher-dimensional space

$$x \Rightarrow z = \varphi(x)$$

$$f(x) = w \cdot \varphi(x) + b$$



z: infinite-dimensional

Kernel trick

$$K(x, x') = \varphi(x) \cdot \varphi(x')$$

$$\int K(x, x')\tilde{\varphi}_i(x')dx' = \lambda_i \tilde{\varphi}_i(x')$$

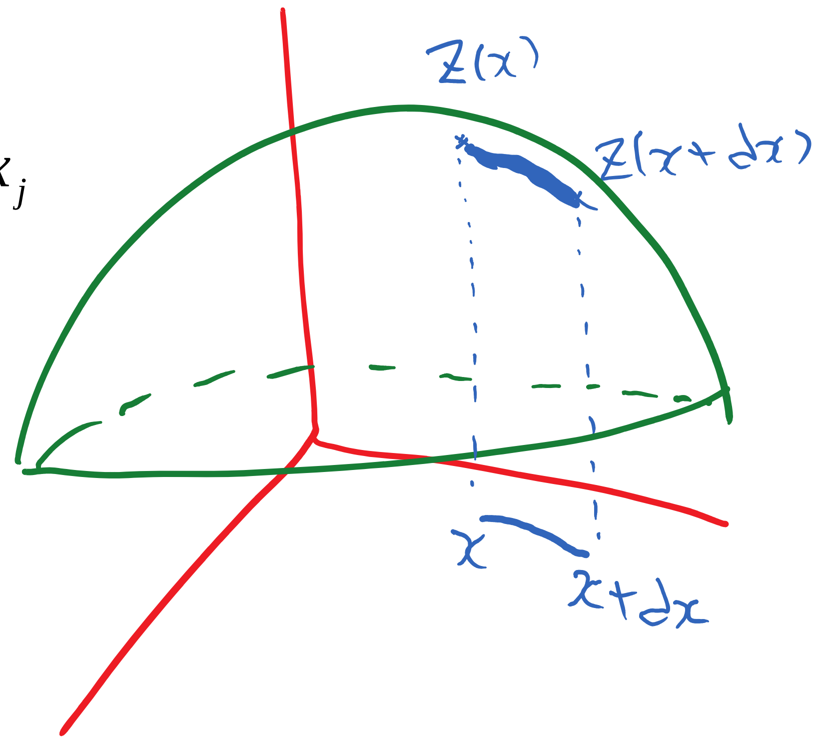
$$\varphi(x) = \frac{1}{\sqrt{\lambda_i}}\tilde{\varphi}_i(x)$$

$$f(x, w) = \sum \alpha_i y_i K(x_i, x)$$

$$ds^2 = | z(x+dx) - z(x) |^2 = \sum \frac{\partial}{\partial x_i} z(x) \frac{\partial}{\partial x_j} z(x) dx_i dx_j$$

$$g_{ij}(x) = \sum \frac{\partial}{\partial x_i} \varphi(x) \frac{\partial}{\partial x_j} \varphi(x)$$

$$g_{ij}(x) = \frac{\partial^2}{\partial x_i \partial x_j} K(x, x')_{|x'=x}$$



Conformal Transformation of Kernel

$$\sigma(x) = \exp\{-\kappa\{f(x)\}^2$$

$$\tilde{K}(x, x') = \sigma(x)\sigma(x')K(x, x')$$

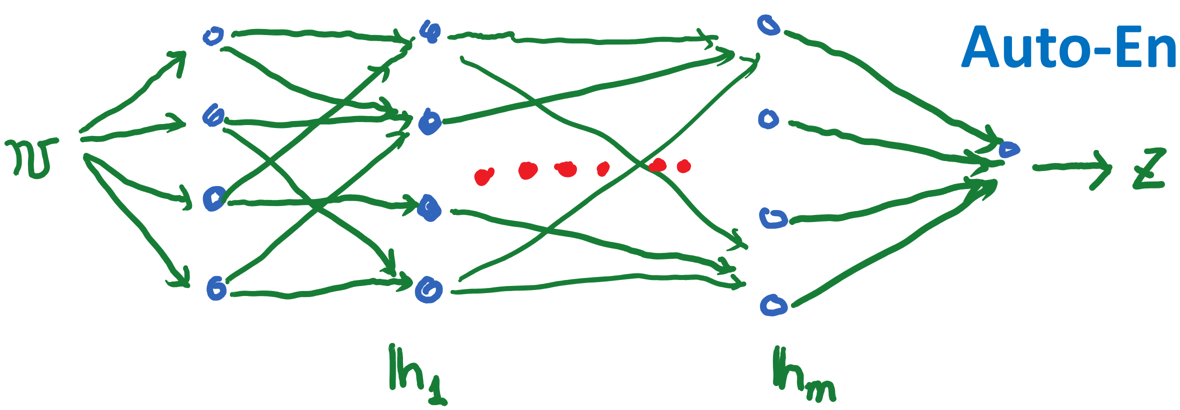
$$\sigma(x) = \sum \exp\{-\kappa_i \,|\, \mathrm{x} - \mathrm{x}_i{}^* |^2\}$$

$$\mathrm{g}_{ij}(x) = \{\sigma(x)\}^2 \, \mathrm{g}_{ij}(x) + \partial_i \sigma(x) \partial_j \sigma(x)$$

Basic Principles of Unsupervised and Supervised Learning Toward Deep Learning

Shun-ichi Amari (RIKEN Brain Science Institute)
collaborators: R. Karakida, M. Okada (U. Tokyo)

Deep Learning



Self-Organization + Supervised Learning

RBM: Restricted Boltzmann Machine
Auto-Encoder, Recurrent Net

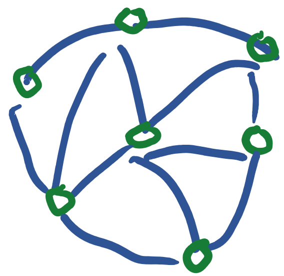
tricks !!
ideas !

Dropout
Contrastive divergence

bi-directional

Boltzmann Machine

Markov chain $\quad P(Z_t \mid Z_{t-1})$



$P(z=1) = f(u)$, $u_i = \sum w_{ij} z_j - b_i$ $\quad w_{ij} = w_{ji}$

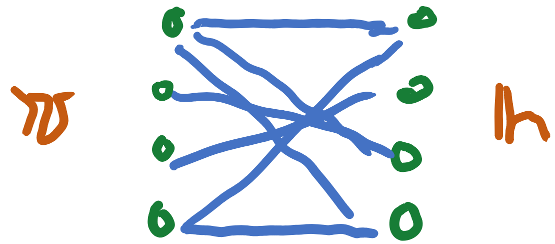
stable distr. $\quad P(Z) = \exp\{\sum s_i z_i + \frac{1}{2}\sum w_{ij} z_i z_j - \psi\}$

RBM: Restricted Boltzmann Machine

$$P(v, h) = \exp\left\{ b \cdot v + c \cdot h + h^T W v - \psi - \frac{1}{2}|v|^2 - \frac{1}{2}|h|^2 \right\}$$

$\sum h_i w_{ij} v_j$

$\sum v_i^2$

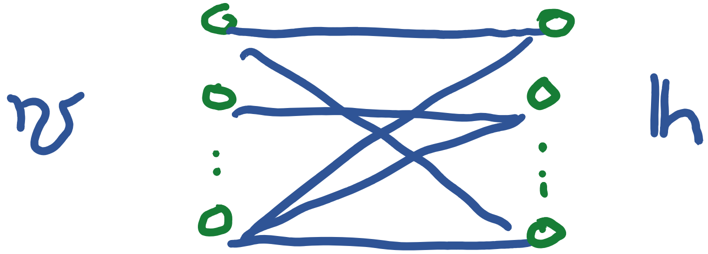


$z = (v, h)$

energy machine :

b, c : neglect

RBM



$$q(v, h) = \exp\{h^T W v - \psi\}$$

marginal distribution :

$$q_v(v) = \sum_h q(v, h) \approx p(v)$$

$$q_h(h) = \sum_v q(v, h)$$

conditional distribution

$$q(h \mid v) = \prod_i f\Big(\sum_j w_{ij} v_j - \tau_i\Big),$$

$$q(v \mid h) = \prod_j f\Big(\sum_i h_i w_{ij} - \tau_j\Big),$$

conditionally independent

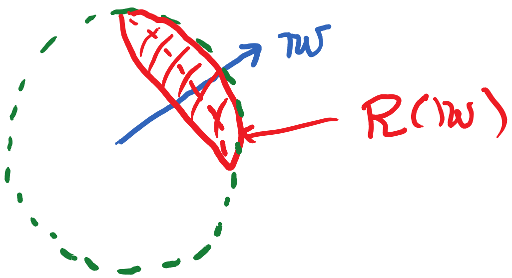
Simple Hebbian Self-Organization

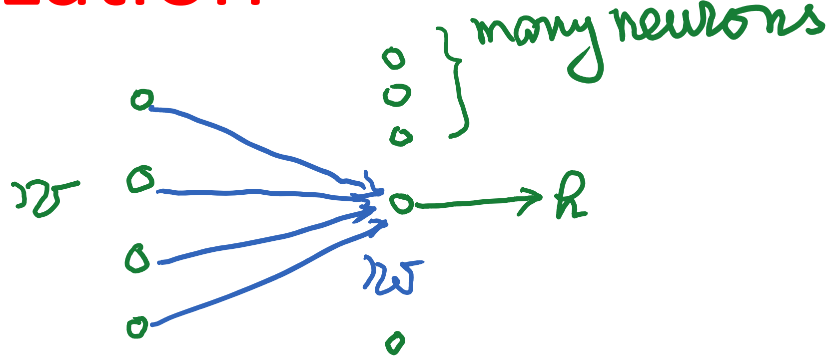
$$h = f(w \cdot v - \tau)$$

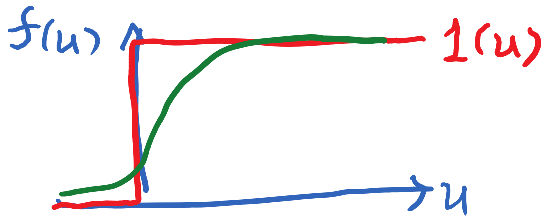
receptive field

$$R(w) = \{ v \mid w \cdot v - v_0 > 0 \}$$

$$|v|^2 = \text{const}$$







self-organization of w : dynamics of $R(w)$

v : $p(v)$

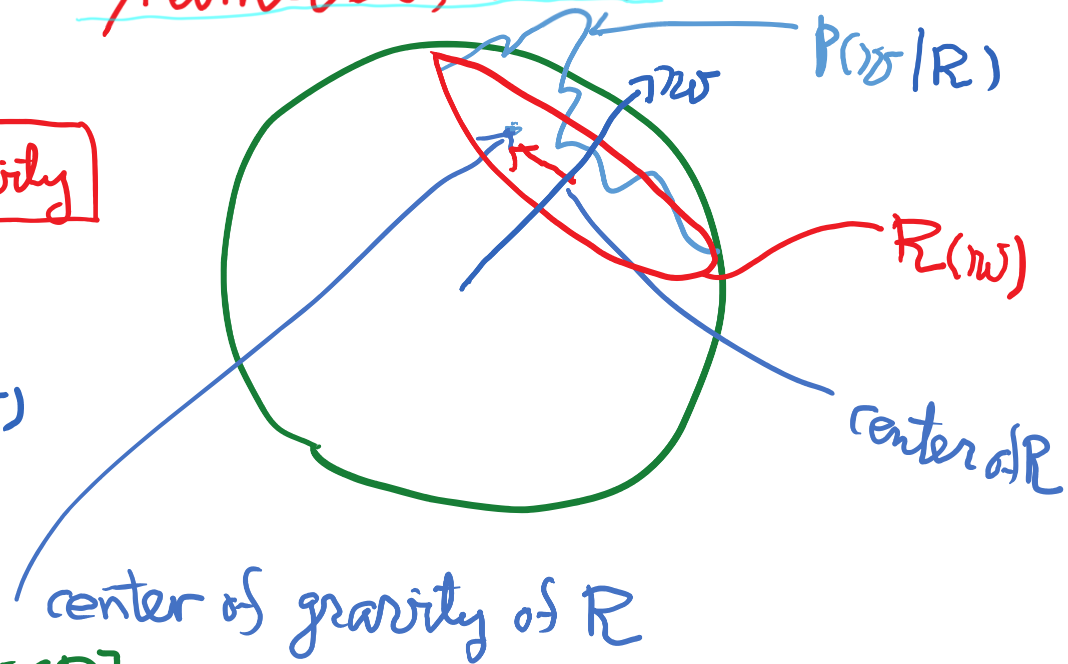
$\boxed{\text{center} \Rightarrow \text{c. of gravity}}$

$$\dot{w} = \langle h(-w + c v)\rangle_{p(v)}$$

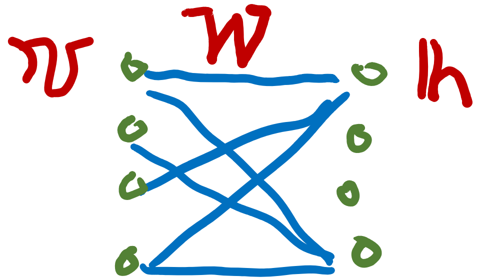
$$\frac{1}{p_R}\dot{w} = -w + c\langle h v\rangle_R$$

$\text{Prob}\{v \in R\} = \langle 1(w \cdot v - v_0)\rangle_{p(v)}$

$E[h v \mid v \in R]$

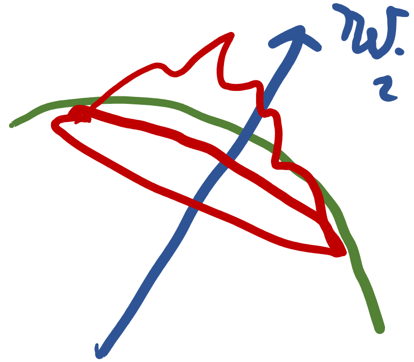


Self-Organization



$$\underset{W}{\text{minimize}}\; KL\left[p(v) : q_v(v\,;W)\right]$$

$$= \int p(v) \log \frac{p(v)}{q_v(v\,;W)}\, dv$$



$$\dot{w}_i = \varepsilon \left\{ -\langle h_i v \rangle_q + \langle h_i v \rangle_p \right\}$$

center w_i ?

center of gravity of $R(w_i)$

$$\langle h v \rangle_q = \frac{\partial}{\partial w} \psi(w)$$

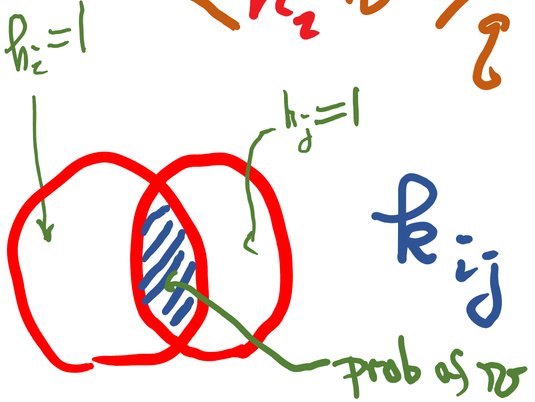
Interaction of Hidden Neurons

$$\langle h v \rangle_q = \frac{\partial \psi(W)}{\partial W} \quad ; \quad q(h, v) = \exp\{ h^T W v \overbrace{- \psi(W)}^{-\frac{1}{2}|v|^2} \}$$

$$\psi(W) = \log \sum_h \int \exp\{ -\frac{1}{2}|v|^2 + h^T W v \} \, dv$$

$$= \log \sum_h \exp\{ \frac{1}{2} |h^T W|^2 \} + c$$

$$\langle h_i v \rangle_q = \sum k_{ij} v_j \quad : \text{interaction}$$



$$k_{ij} = E_q[h_i h_j] \leftarrow \text{joint firing prob}$$

v: analog, Gaussian

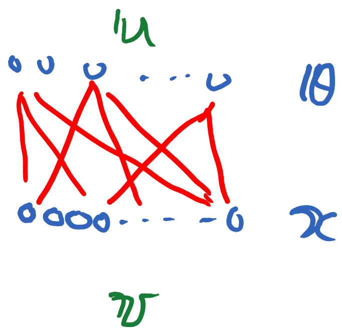
Bayesian Duality in Exponential Family

Data x **Parameter (higher-order concepts)** θ

$$p(x\mid\theta) = \exp\{\theta\cdot x - \bar{k}(x) - \psi(\theta)\}$$

$$p(x,\theta) = \exp\{\theta\cdot x - \bar{k}(x) - \bar{\psi}(\theta)\} \quad \psi - \log\pi(\theta)$$

$$p(\theta\mid x) = \exp\{\theta\cdot x - \bar{k}(x) - \bar{\psi}(\theta)\}$$



Curved exponential family

$$x(v),\quad \theta(u)\ :\ \exp\{\theta(u)\cdot x(v) - \bar{k} - \bar{\psi}\}$$

RBM : curved exp. family

$\theta = h, \quad x = Wv$

$x = v \quad \theta = hW$

$$P(v, h) = \exp\{ \theta \cdot x - \bar{k} - \bar{\psi} \}$$

$$\theta \cdot x = h^T W v$$

Gaussian Boltzmann Machine

$$q(v, h) = \exp\left\{-\frac{1}{2}|v|^2 - \frac{1}{2}|h|^2 + h^T W v - \psi(W)\right\}$$

$$\langle h v \rangle_p = W C \qquad C = \int v v^T p(v) dv$$

center of gravity

covariance matrix

$$\langle h v \rangle_q = (I - WW^T)^{-1} W$$

interactions of h neurons

Equilibrium Solution

$$WC = (I - WW^T)^{-1} W \qquad W = \begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix}$$

solution : $w_i \cdot w_j = 0 \quad (i \neq j), \quad |w_i|^2 = m_i$

$$w_i C = \underbrace{(1 - m_i)^{-1}}_{\lambda_i} w_i \qquad w_i C = \lambda_i w_i$$

PCA!

$$m_i = 1 - \frac{1}{\lambda_i}$$

Equilibrium Solution

$$WC = (I - WW^T)^{-1}W$$

General Solution

$$W = U\mathrm{diag}\left(\sqrt{1-\frac{1}{\lambda_1}}, \ldots, \sqrt{1-\frac{1}{\lambda_m}}, 0, \ldots, 0\right)V$$

- othogonal matrix : U, V
- C diagonalized by V $\quad C = V^T\mathrm{diag}(\lambda_1, \ldots, \lambda_n)V$

You can choose m(≤ k) eigen values form $\lambda_1, \ldots, \lambda_k > 1$

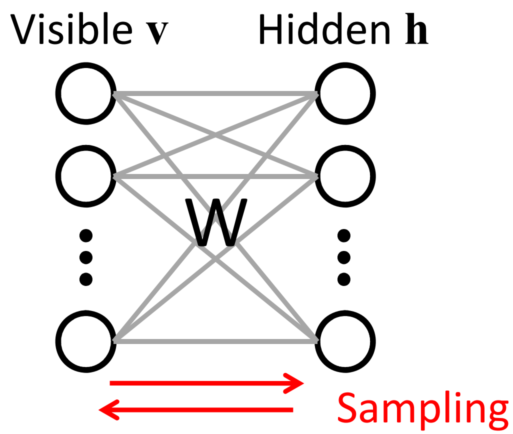
Stable Solution

the case of m = k

Contrastive Divergence

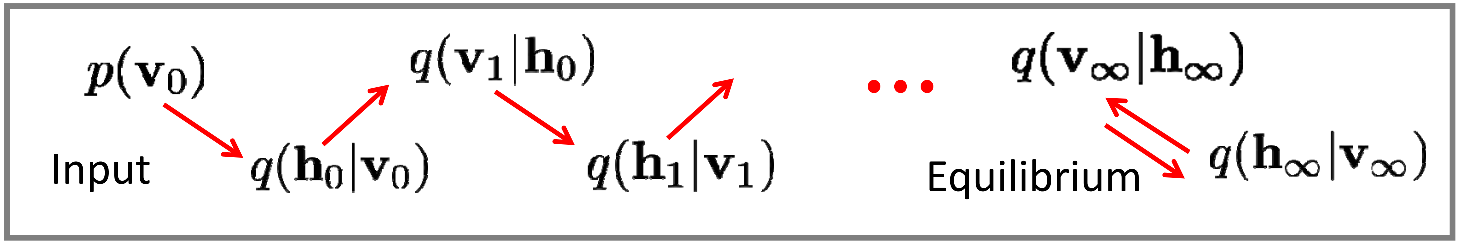
- RBM
  - 2-layered probabilistic neural network
  - No connections within layers

$$q(\mathbf{h},\mathbf{v};W) = \exp\left(-\mathbf{h}^T W \mathbf{v}\right) / \sum_{\mathbf{h},\mathbf{v}} \exp\left(-\mathbf{h}^T W \mathbf{v}\right)$$



- How to train RBM

  Maximum Likelihood (ML) learning is hard



Many iterations of Gibbs Sampling demand too much computational time

Contrastive Divergence Solution

$$\langle h v \rangle_{\tilde{q}} = W C W^T W + W$$

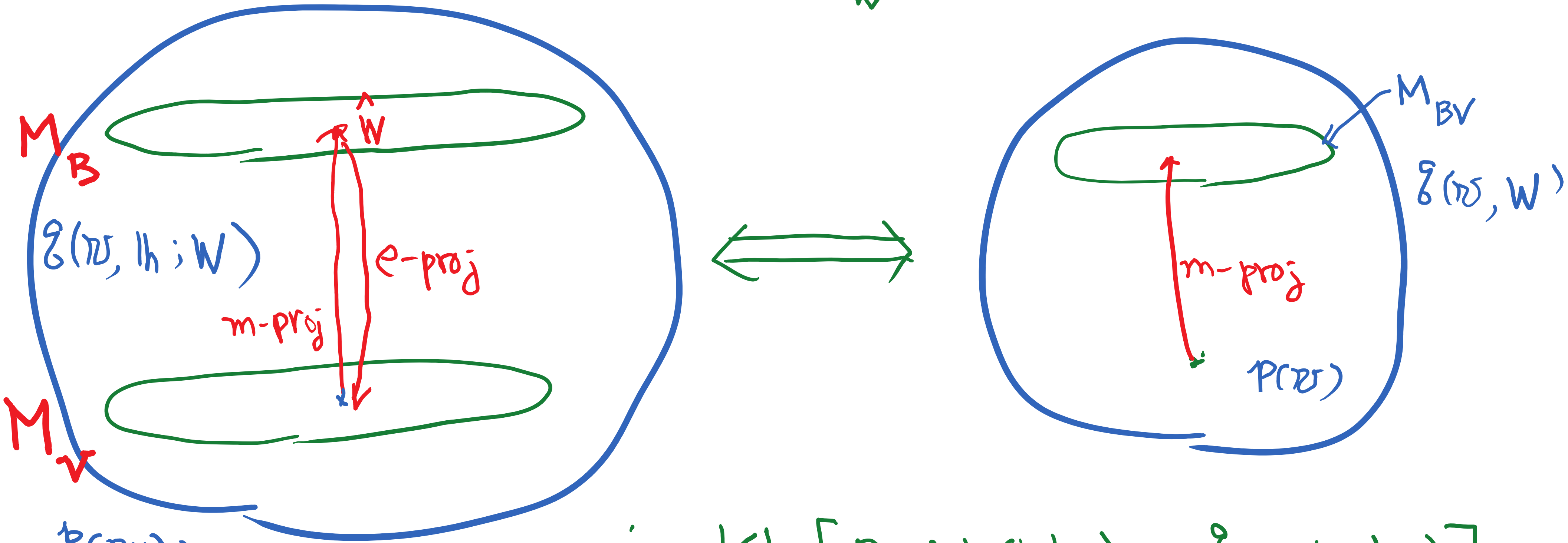
$$W C = W C W^T W + W$$

$$w_i C = \frac{1}{1 - m_i} w_i$$

PCA!

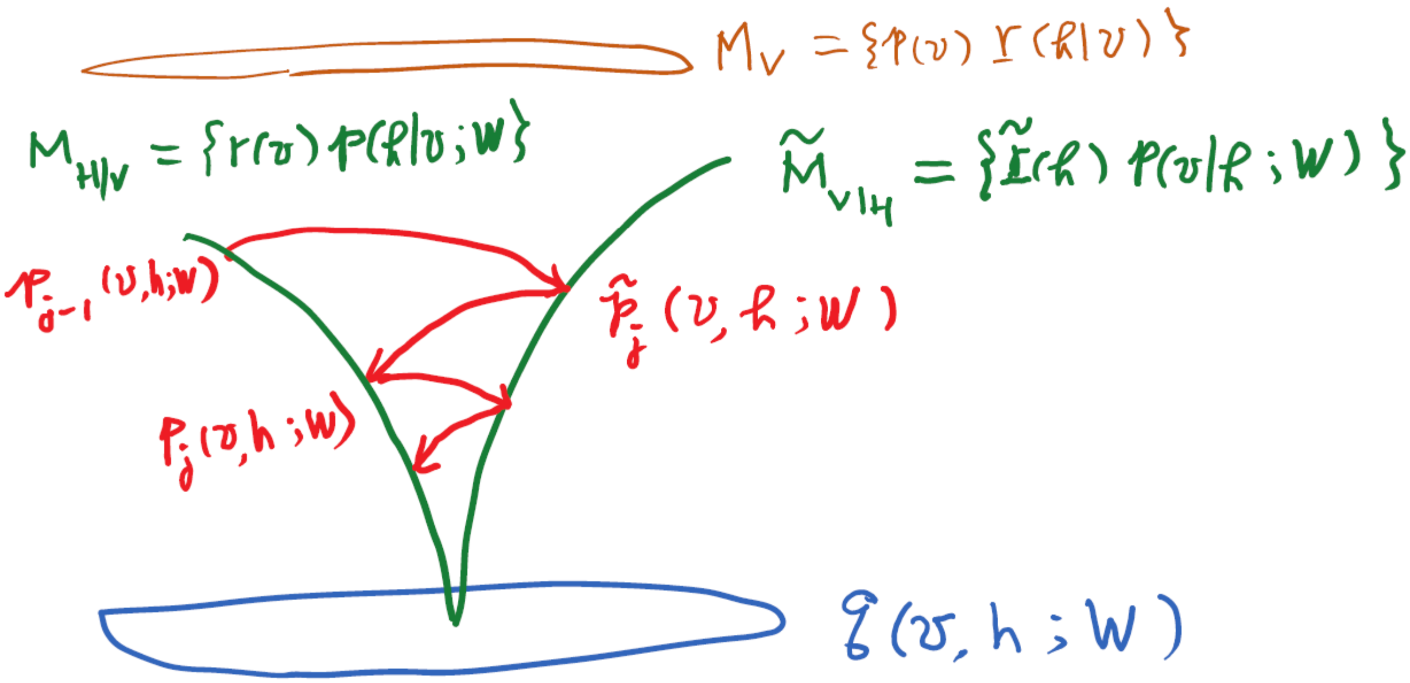
Two Manifolds

$$\min_{W} KL[p(v) : q_V(v;W)]$$



$$\min_{r;W} KL[p(v)r(h|v) : q(v,h;W)]$$

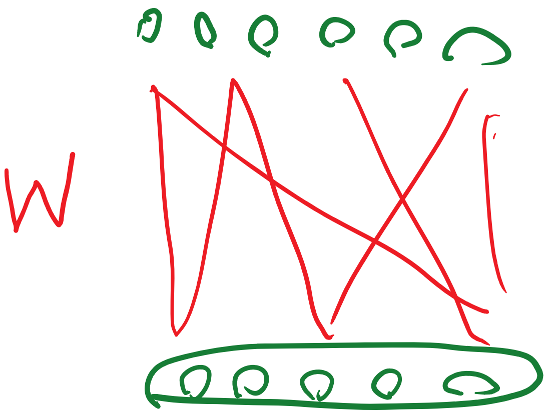
Geometry of CDn (contrastive divergence)



Bernoulli-Gaussian RBM

ICA

R. Karakida



binary h

Gaussian v

$p(v)$: $v = \underline{OD} s$

$W = A^{-1}$

Simulation

The number of Neurons: N = M = 2, σ = 1/2

Sources p (s)

Uniform Distribution

Mixing

Input

$$\mathbf{v} = B\mathbf{s}$$

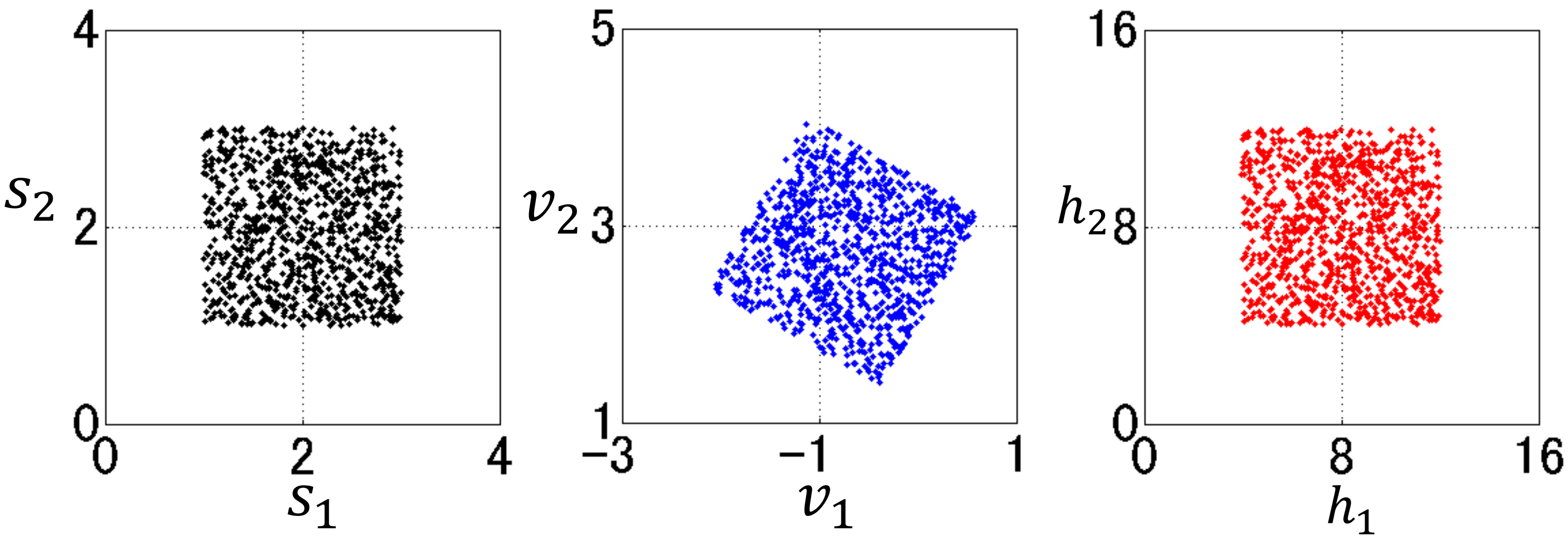
CD_1

ICA Solution

$$\bar{W} = DB^T$$

Output

$$\mathbf{h} = \bar{W}\mathbf{v}$$



Independent sources are extracted in G-B RBM

Information Geometry of Neuromanifolds

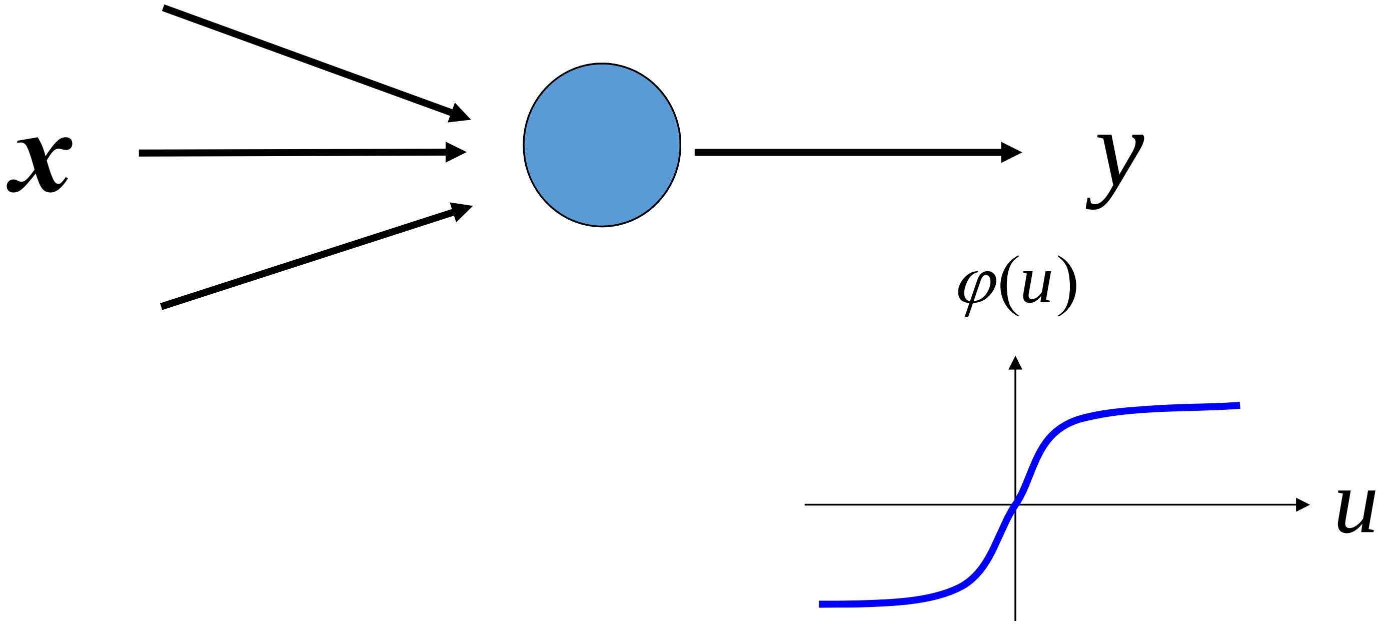
Natural Gradient and Singularities

Multilayer Perceptron

Shun-ichi Amari
RIKEN Brain Science Institute

Mathematical Neurons

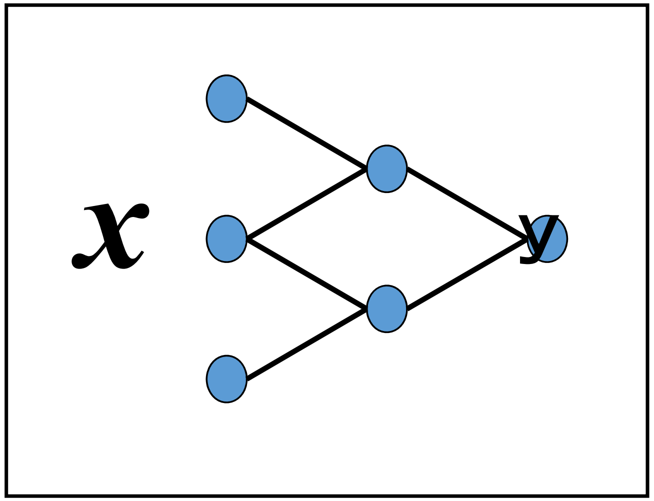
$$y = \varphi\left(\sum w_i x_i - h\right) = \varphi\left(\boldsymbol{w} \cdot \boldsymbol{x}\right)$$



Multilayer Perceptrons

$$y = \sum v_i \varphi\left(\boldsymbol{w}_i \cdot \boldsymbol{x}\right) + n$$

$$x = \left(x_1, x_2, \ldots, x_n\right)$$



$$p\left(y \middle| \boldsymbol{x}; \boldsymbol{\theta}\right) = c \exp\left\{-\frac{1}{2}\left(y - f\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)^2\right\}$$

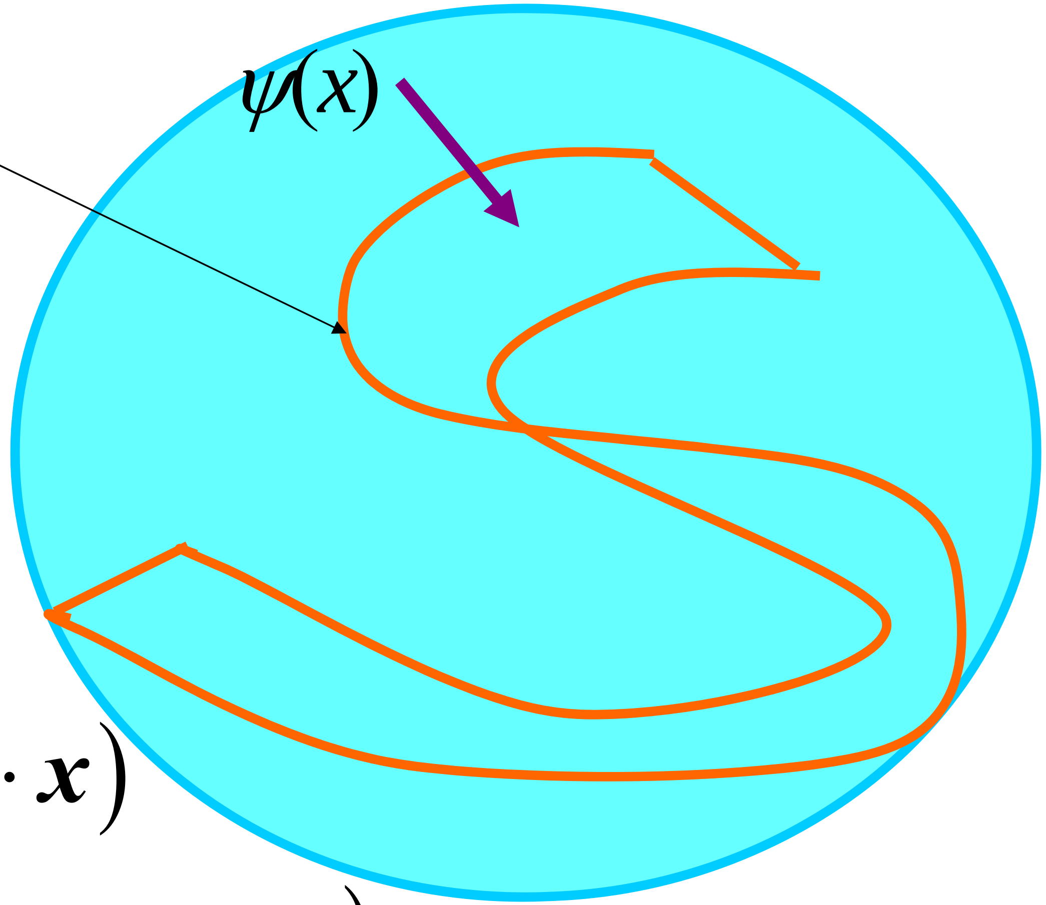
$$f\left(\boldsymbol{x}, \boldsymbol{\theta}\right) = \sum v_i \varphi\left(\boldsymbol{w}_i \cdot \boldsymbol{x}\right)$$

$$\theta = \left(w_1, \ldots, w_m; v_1, \ldots, v_m\right)$$

Multilayer Perceptron: Neuromanifold

neuromanifold

space of functions



$$y = f(\boldsymbol{x}, \boldsymbol{\theta})$$

$$= \sum v_i \, \varphi\left(\boldsymbol{w}_i \cdot \boldsymbol{x}\right)$$

$$\boldsymbol{\theta} = \left(v_1 \cdots, v_m \; ; \; \boldsymbol{w}_{1,} \cdots, \boldsymbol{w}_m\right)$$

Universal Function Approximator

$$\psi(\boldsymbol{x}) \approx \sum_{i=1}^{N} \mathrm{v}_{\mathrm{i}} a_i(\boldsymbol{x})$$

$$\psi(\boldsymbol{x}) \approx \sum_{i=1}^{m} v_i \varphi_i(\boldsymbol{x})$$

$$\varphi_i(\boldsymbol{x}) = \varphi(\boldsymbol{w}_i \cdot \boldsymbol{x})$$

Learning from examples

$$\psi(\boldsymbol{x}) \approx f\left(x, \hat{\theta}\right)$$

examples $\cdots$ $D = \left\{\left(\boldsymbol{x}_1,\ y_1\right),\ \cdots,\ \left(\boldsymbol{x}_n,\ y_n\right)\right\}$

learning ; estimation

Backpropagation ---gradient learning

$$\text{examples}:(y_1,\boldsymbol{x}_1),\cdots(y_t,\boldsymbol{x}_t)\text{--training set}$$

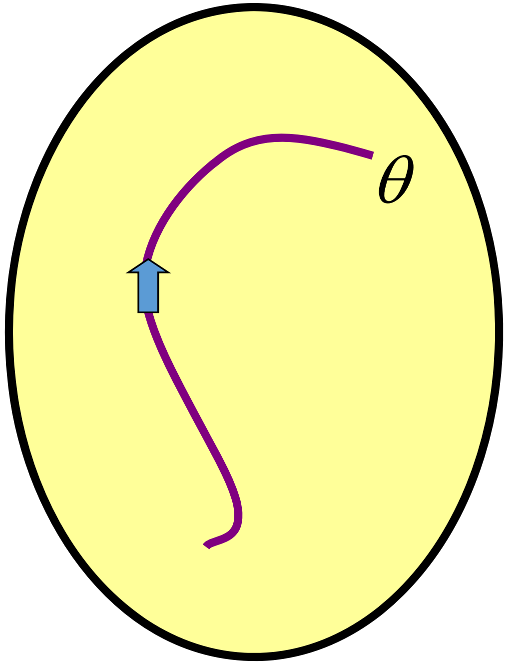
$$y = f(x,\theta)+n$$

$$E(y,x;\theta)=\frac{1}{2}\left|y-f(\boldsymbol{x},\boldsymbol{\theta})\right|^2$$

$$=-\log p(y,\boldsymbol{x};\boldsymbol{\theta})$$

$$\Delta\boldsymbol{\theta}_t=-\eta_t\frac{\partial E}{\partial\boldsymbol{\theta}}$$

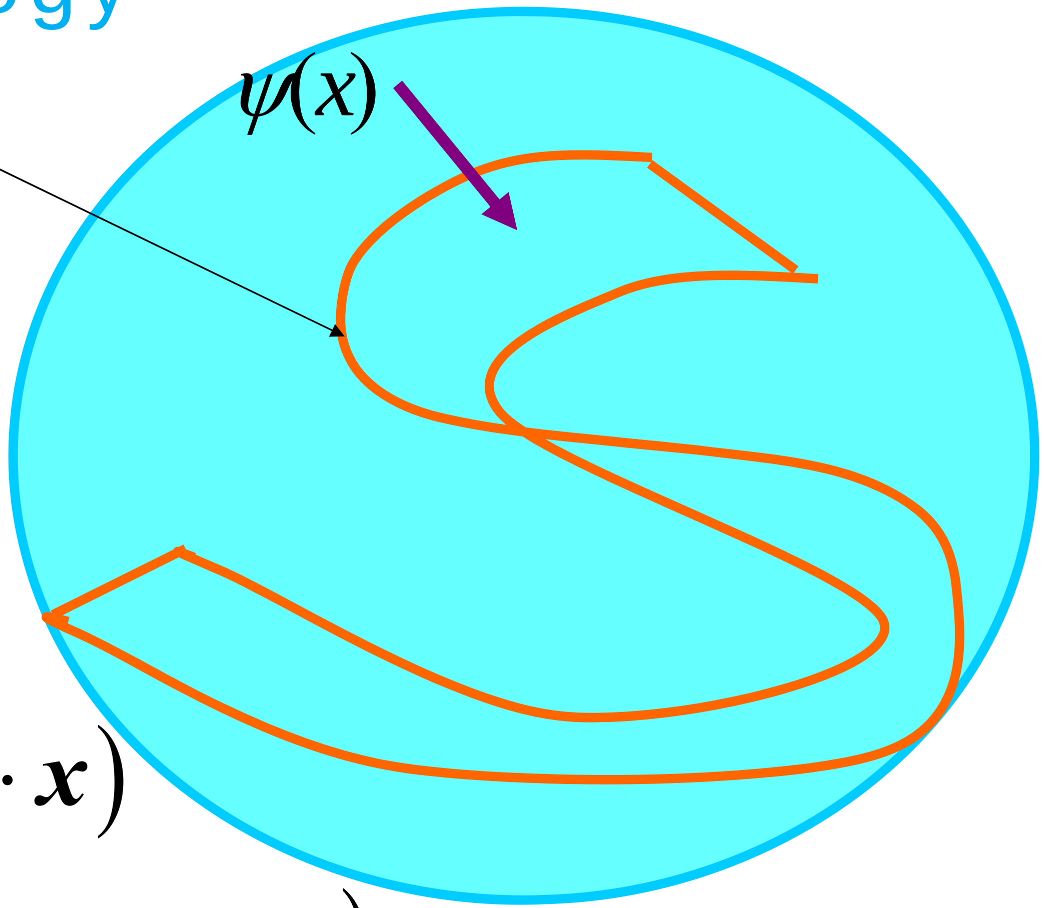
$$f(\boldsymbol{x},\boldsymbol{\theta})=\sum v_i\varphi(\boldsymbol{w}_i\cdot\boldsymbol{x})$$



Multilayer Perceptron: Neuromanifold
Metric and Topology

neuromanifold

space of functions



$$y = f(\boldsymbol{x}, \boldsymbol{\theta})$$

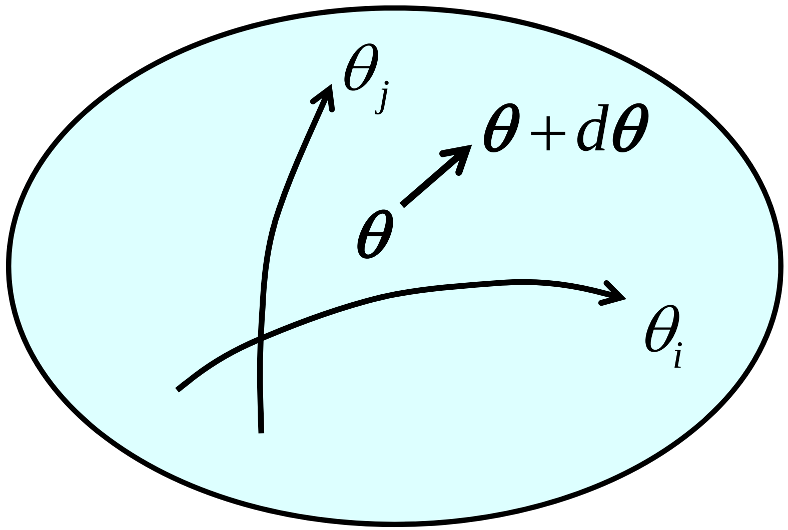
$$= \sum v_i \, \varphi \left(\boldsymbol{w}_i \cdot \boldsymbol{x} \right)$$

$$\boldsymbol{\theta} = \left(v_1 \cdots, v_m \; ; \; \boldsymbol{w}_{1,} \cdots, \boldsymbol{w}_m \right)$$

Metric: Riemannian manifold

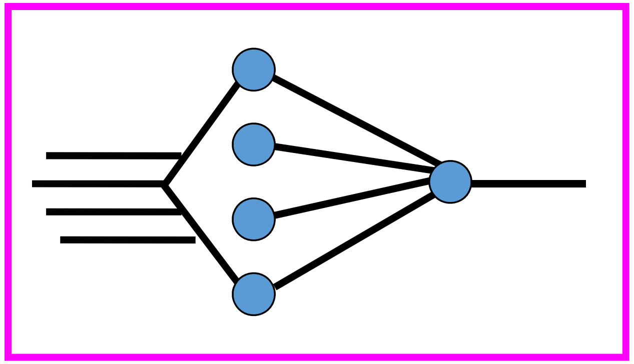
$$g_{ij}(\theta) = E[\frac{\partial \log p(y \mid x;\theta)\partial \log p(y \mid x;\theta)}{\partial\theta_i \partial\theta_j}]$$

$$\begin{aligned} ds^2 &= \left|d\boldsymbol{\theta}\right|^2 \\ &= \sum g_{ij}\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta}_i d\boldsymbol{\theta}_j \\ &= d\boldsymbol{\theta}^T G\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta} \end{aligned}$$

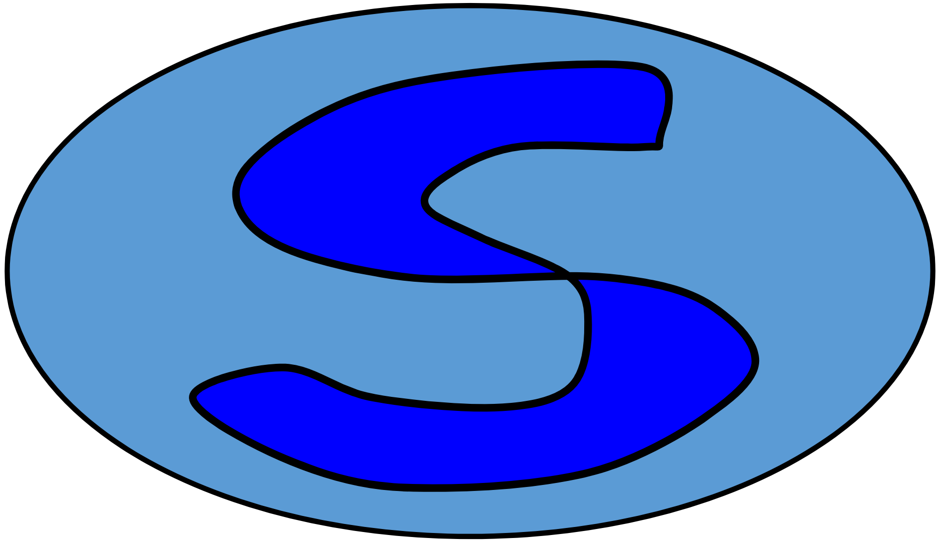


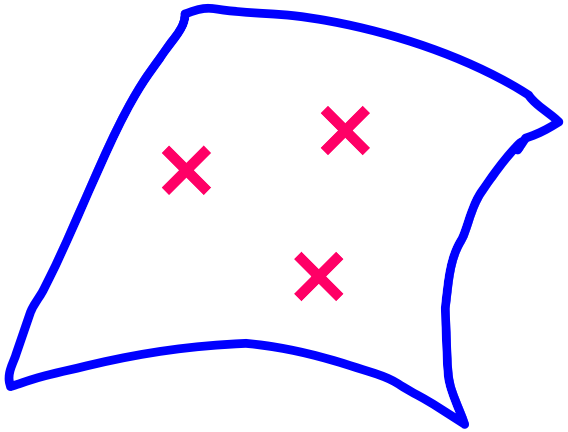
Topology: Neuromanifold

- Metrical structure
- Topological structure

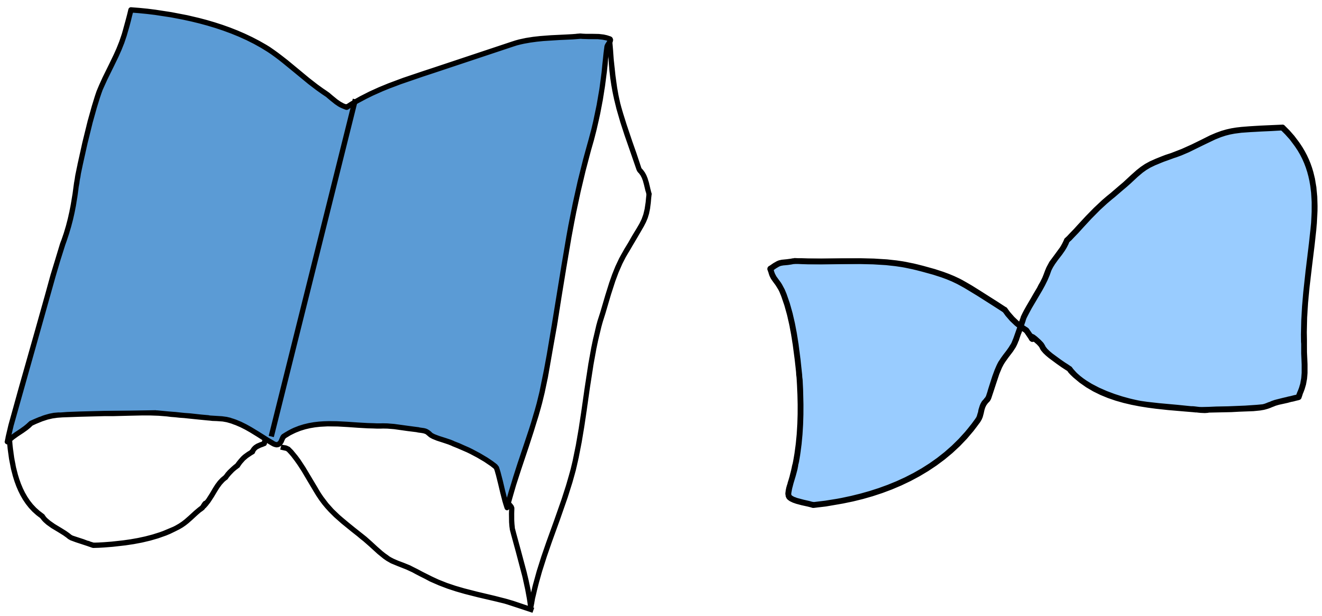


θ





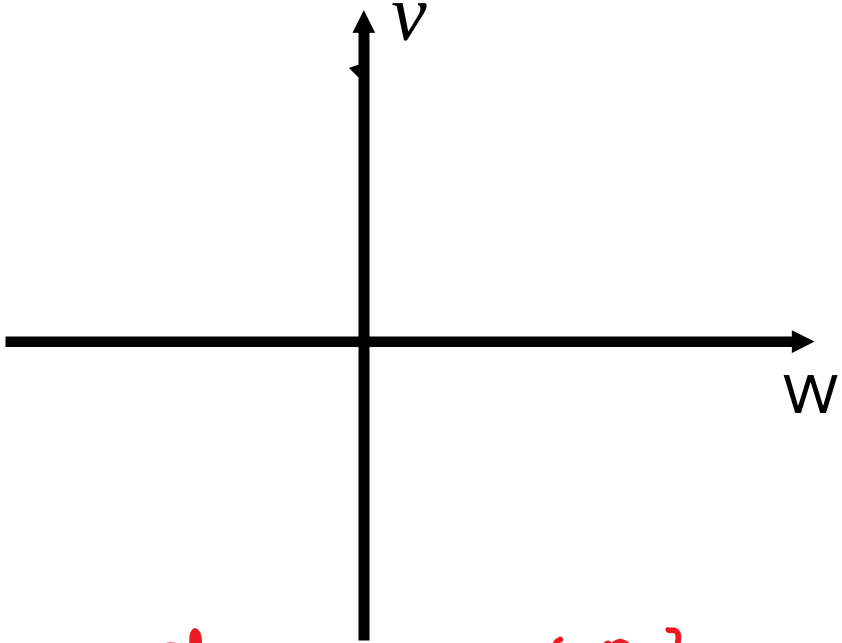
singularities



Geometry of singular model

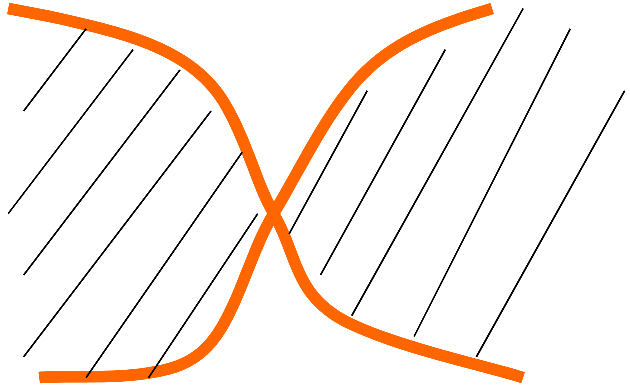
$$y = v\varphi(\boldsymbol{w}\cdot\boldsymbol{x}) + n$$

$$v\,|\,\mathbf{w}\,| = 0$$



parameter space $\{\theta\}$

$\Rightarrow$

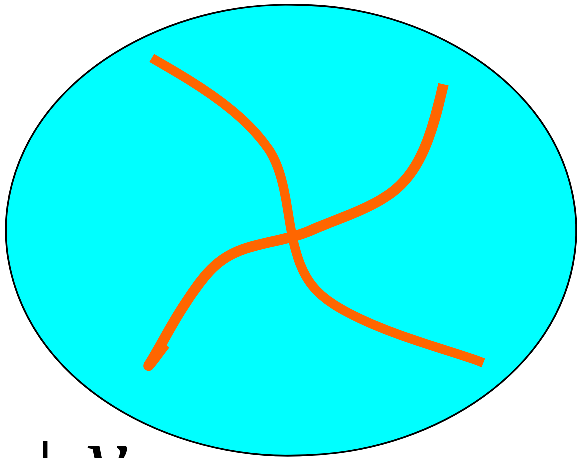


behavioral space $\{f(x,\theta)\}$

Parameter Space S

$$S = \{\boldsymbol{\theta}\} \qquad y = \sum v_i \varphi(\boldsymbol{w}_i \cdot \boldsymbol{x}) + n$$

Equivalence



1) $v_i \boldsymbol{w}_i = 0$

2) $\boldsymbol{w}_i = \boldsymbol{w}_j \Rightarrow v_i + v_j$

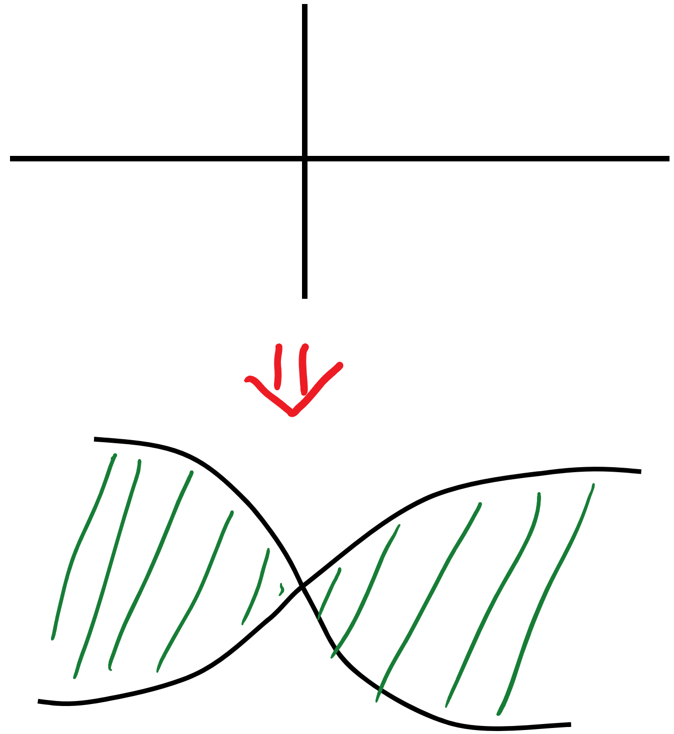
$$M = S / \approx$$

Topological Singularities

S : parameter space

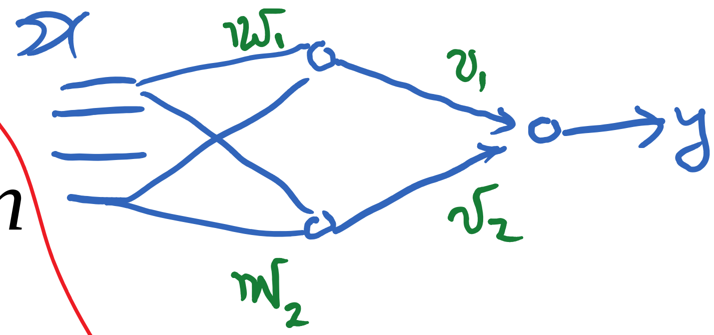
$$M = S/\approx$$

M : behavioral space



2 hidden-units

$$y = v_1 \varphi\left(\boldsymbol{w}_1 \cdot \boldsymbol{x}\right) + v_2 \varphi\left(\boldsymbol{w}_2 \cdot \boldsymbol{x}\right) + n$$

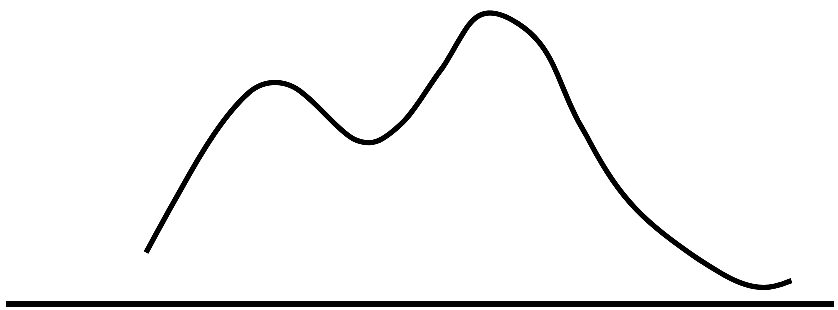


$$S:\ v_1 v_2 \left|\boldsymbol{w}_1 - \boldsymbol{w}_2\right| \left|\boldsymbol{w}_1 + \boldsymbol{w}_2\right| = 0$$

$$\left(1 - v\right)\varphi\left(x - w_1\right) + v\varphi\left(x - w_2\right)$$

Gaussian mixtures

$$p(x) = \sum v_i \exp\left\{-\frac{1}{2}\left(x - w_i\right)^2\right\}$$

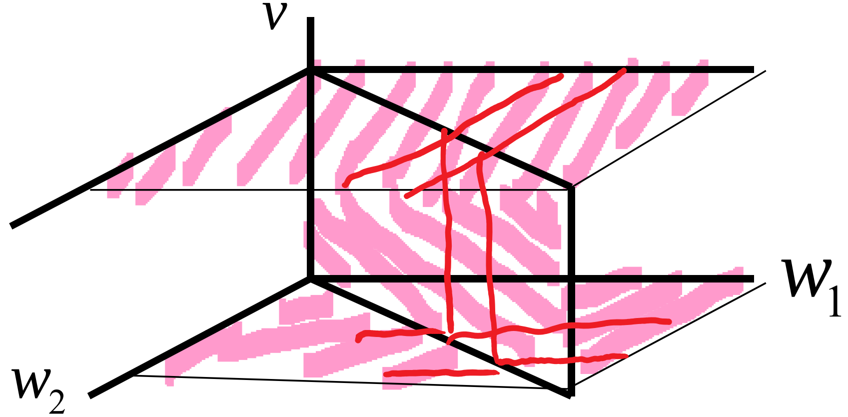


Gaussian mixture

$$p(x; v, w_1, w_2) = (1-v)\varphi(x-w_1) + v\varphi(x-w_2)$$

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\}$$

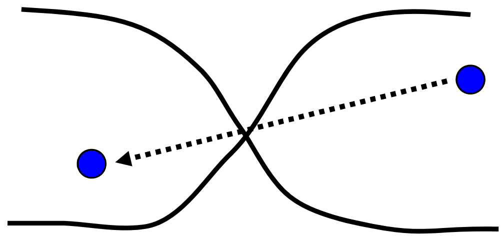
singular: $w_1 = w_2$, $v(1-v) = 0$



Learning, Estimation, and Model Selection

$$E_{\text{gen}} = D\left[p_0\left(y \middle| \boldsymbol{x}\right) : p\left(y \middle| \boldsymbol{x}; \hat{\boldsymbol{\theta}}\right)\right]$$

$$E_{\text{train}} = D\left[p_{\text{emp}}\left(y \middle| \boldsymbol{x}; \hat{\boldsymbol{\theta}}\right)\right]$$



$$E_{\text{gen}} = \frac{d}{2n}$$

d : dimension

$$E_{\text{gen}} = E_{\text{train}} + \frac{d}{n}$$

$\frac{\log n}{n}$, $\frac{\log \log n}{n}$, ...

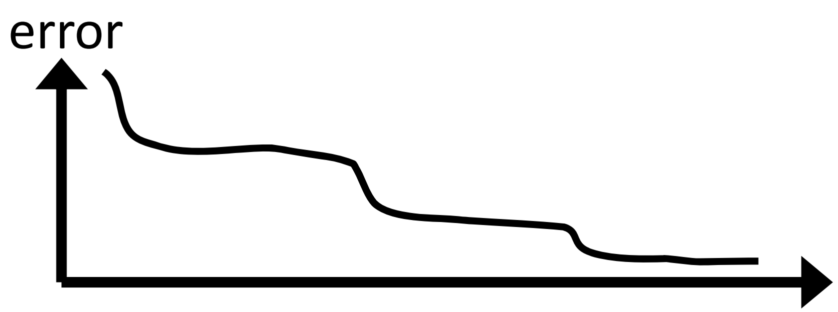
Problem of Backprop

$$\Delta \boldsymbol{\theta}_t = -\eta_t \nabla l(x_t, y_t; \theta_t)$$

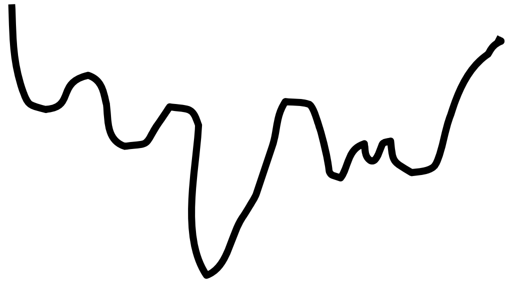
- **slow convergence----plateau---saddle**
- **local minima**

Flaws of MLP

slow convergence : plateau

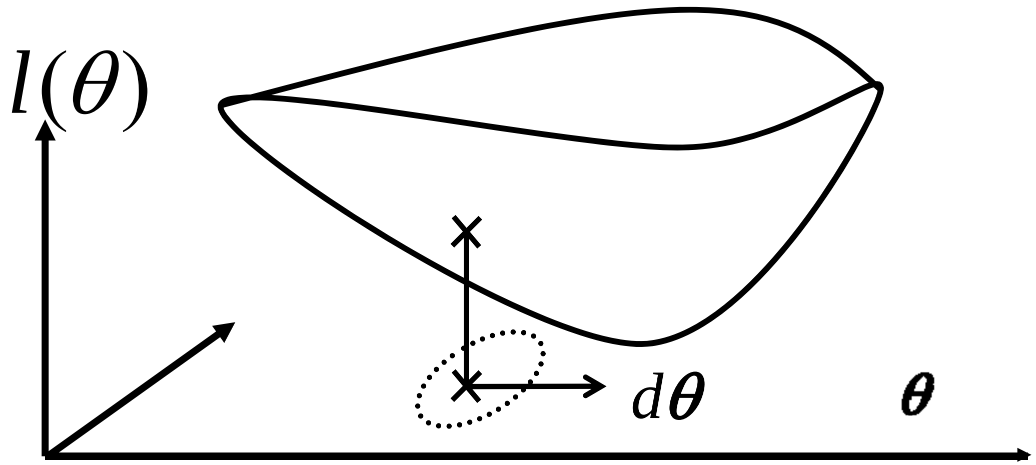


local minima



➡ Boosting, Bagging, SVM

Steepest Direction --- Natural Gradient



$$\nabla l = \left(\frac{\partial l}{\partial \theta_1}, \cdots, \frac{\partial l}{\partial \theta_n} \right)$$

$$\Delta \boldsymbol{\theta}_t = -\eta_t \nabla l(x_t, y_t; \theta_t)$$

$$\widetilde{\nabla} l = G^{-1}(\boldsymbol{\theta}) \nabla l$$

$$|d\boldsymbol{\theta}|^2 = d\boldsymbol{\theta}^T G d\boldsymbol{\theta} = \sum G_{ij} d\theta^i d\theta^j$$

Natural Gradient

$$\max \qquad dl = l(\boldsymbol{\theta} + d\boldsymbol{\theta}) - l(\boldsymbol{\theta}) = \nabla l \cdot d\theta$$

$$\text{under } \left|d\boldsymbol{\theta}\right|^2 = \sum g_{ij} d\theta_i d\theta_j = \varepsilon^2$$

$$d\theta \approx \tilde{\nabla} l = G^{-1}(\boldsymbol{\theta}) \nabla l$$

$$\Delta\boldsymbol{\theta}_t = -\eta_t \tilde{\nabla} l(x_t, y_t; \theta_t)$$

Information Geometry of MLP

Natural Gradient Learning :
S. Amari ; H.Y. Park

$$\Delta\boldsymbol{\theta} = -\eta G^{-1}(\boldsymbol{\theta})\frac{\partial l}{\partial \boldsymbol{\theta}}$$

Adaptive natural gradient learning

$$G_{t+1}^{-1} = (1+\varepsilon)G_t^{-1} - \varepsilon G_t^{-1}\nabla f \nabla f^T G_t^{-1}$$

Regular statistical model

$$M = \{ p(x, \theta) \}$$

G : Fisher information

$$E\left[\Delta\theta \Delta\theta^T \right] = \frac{1}{n} G^{-1}$$

$$E\left[KL\left[p(x, \theta_0) : p\left(x, \hat{\theta}\right) \right] \right] \approx \frac{1}{2n} G \cdot E[\Delta\theta \Delta\theta]$$

$$\approx \frac{d}{2n}$$

AIC, MDL

Landscape of error at singularity

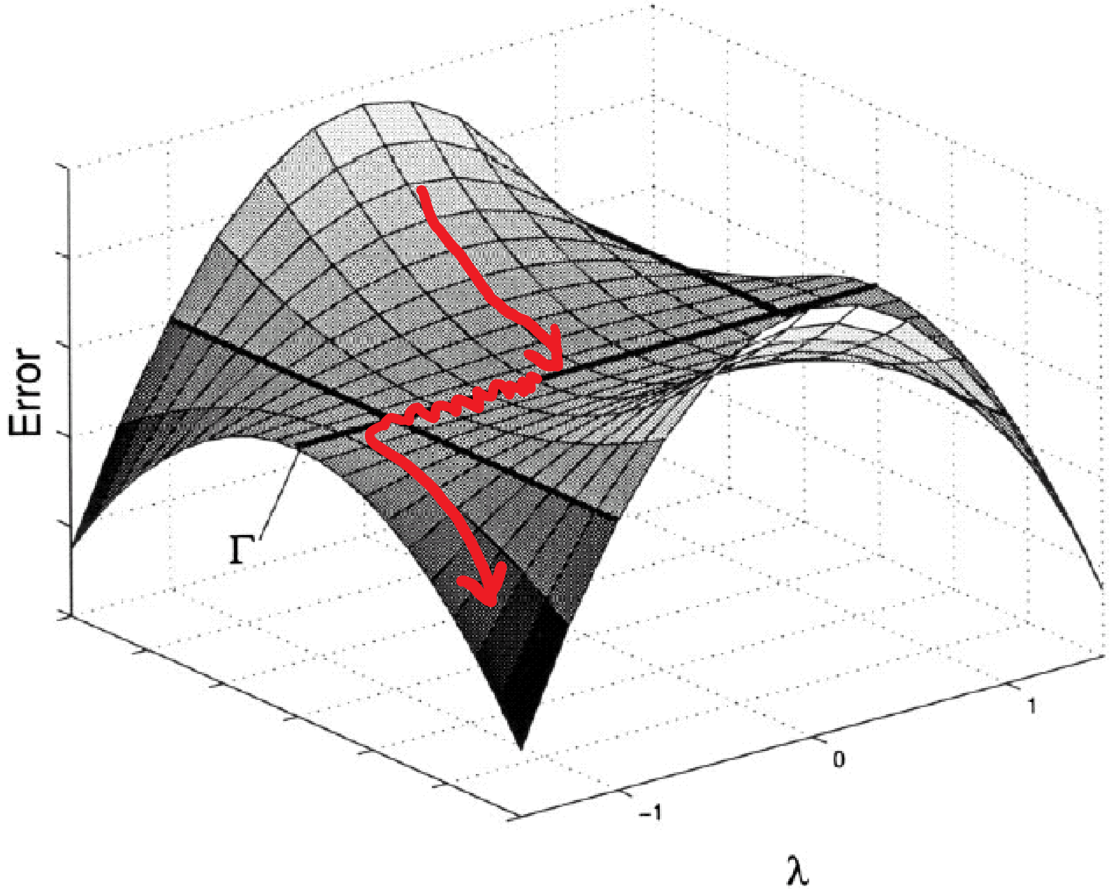
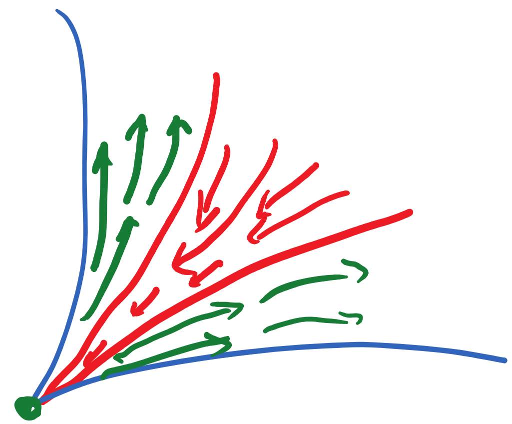


Fig. 5. Critical set with local minima and plateaus.

Milner attractor



Dynamics of Learning

$$\frac{d\theta}{dt} = -\eta \nabla l, \qquad \frac{d\theta}{dt} = -\eta G^{-1} \nabla l$$

$$\frac{du}{dt} = f(u,z), \qquad \frac{dz}{dt} = k(u,z)$$

$$\frac{du}{dz} = \frac{f(u,z)}{k(u,z)}, \qquad u^2 = z^2 - \frac{1}{2}\log|z| + c$$

Coordinate Transformation

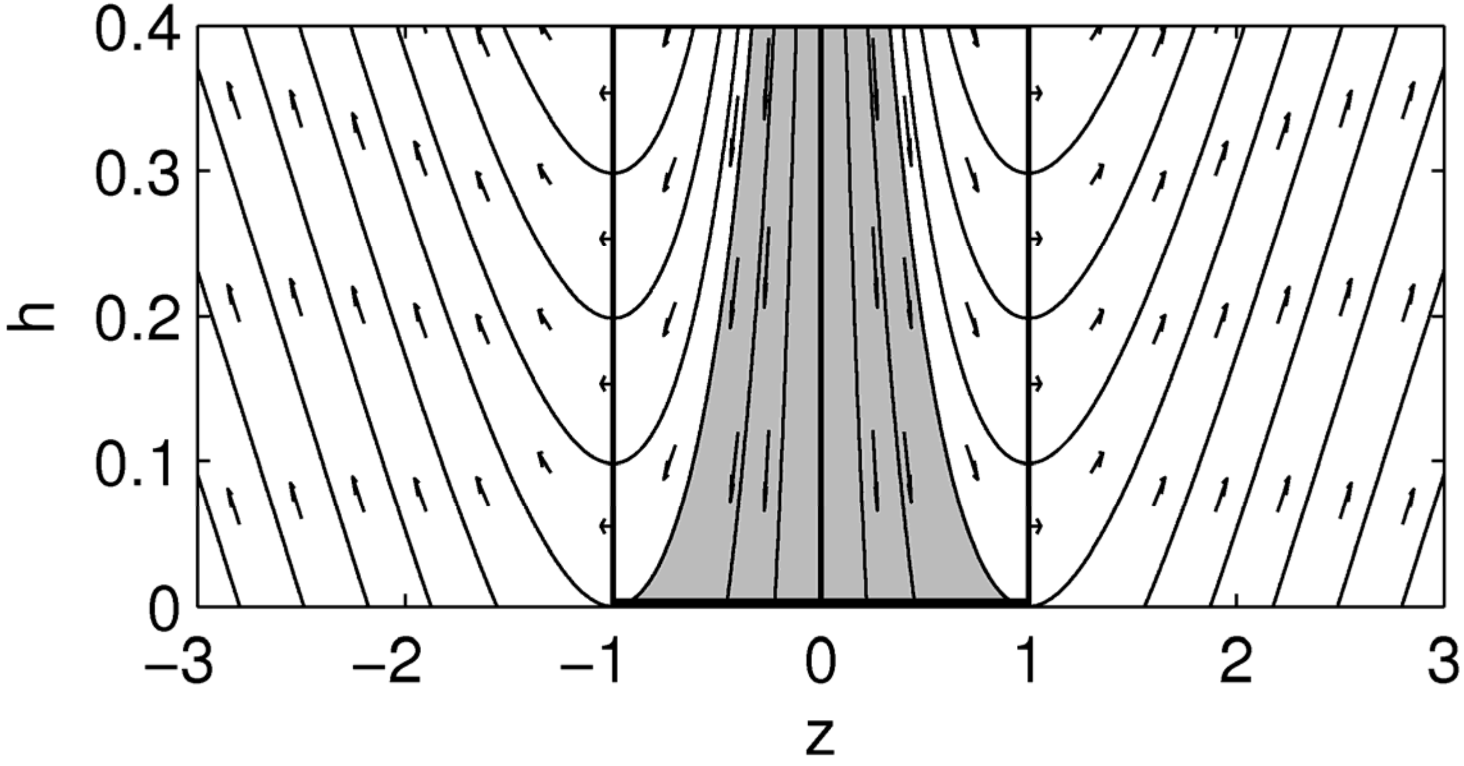
$$
\begin{cases}
\boldsymbol{u} = \boldsymbol{w}_2 - \boldsymbol{w}_1 & : \boldsymbol{u} = 0 \qquad \mathcal{R}_1 \\
\boldsymbol{w} = \dfrac{v_1 \boldsymbol{w}_1 + v_2 \boldsymbol{w}_2}{v} & \boldsymbol{w} = \boldsymbol{w}^* \\
v = v_1 + v_2 & v = v^* \\
z = \dfrac{v_2 - v_2}{v} & z = \pm 1 \qquad \mathcal{R}_2
\end{cases}
$$

Dynamics of Learning: Natural gradient works!!

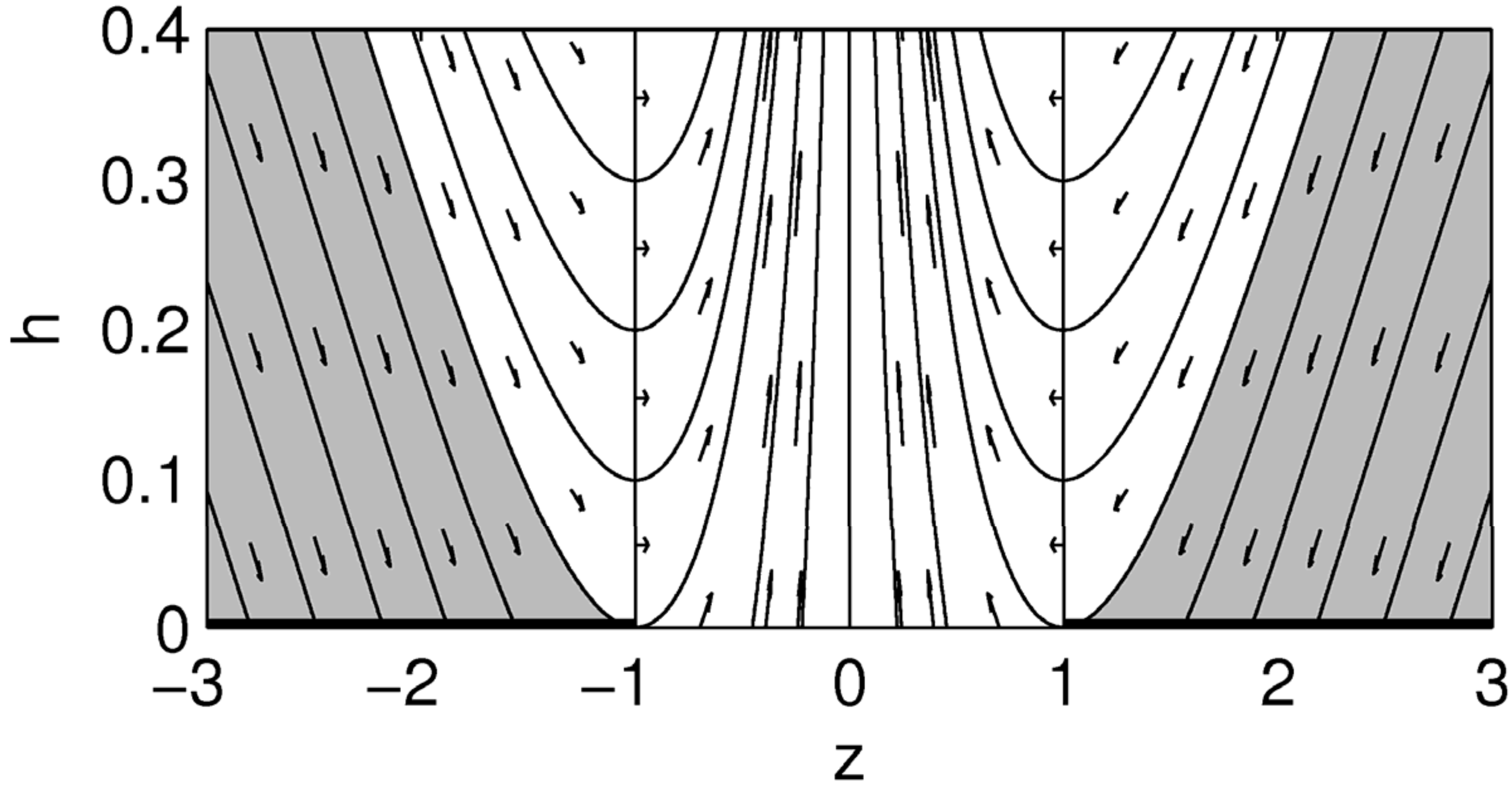
$$\frac{d\theta}{dt} = -\eta \nabla l, \qquad \frac{d\theta}{dt} = -\eta G^{-1} \nabla l$$

$$\frac{du}{dt} = f(u, z), \quad \frac{dz}{dt} = k(u, z)$$

$$\frac{du}{dz} = \frac{f(u, z)}{k(u, z)}, \qquad u^2 = z^2 - \frac{1}{2}\log|z| + c$$

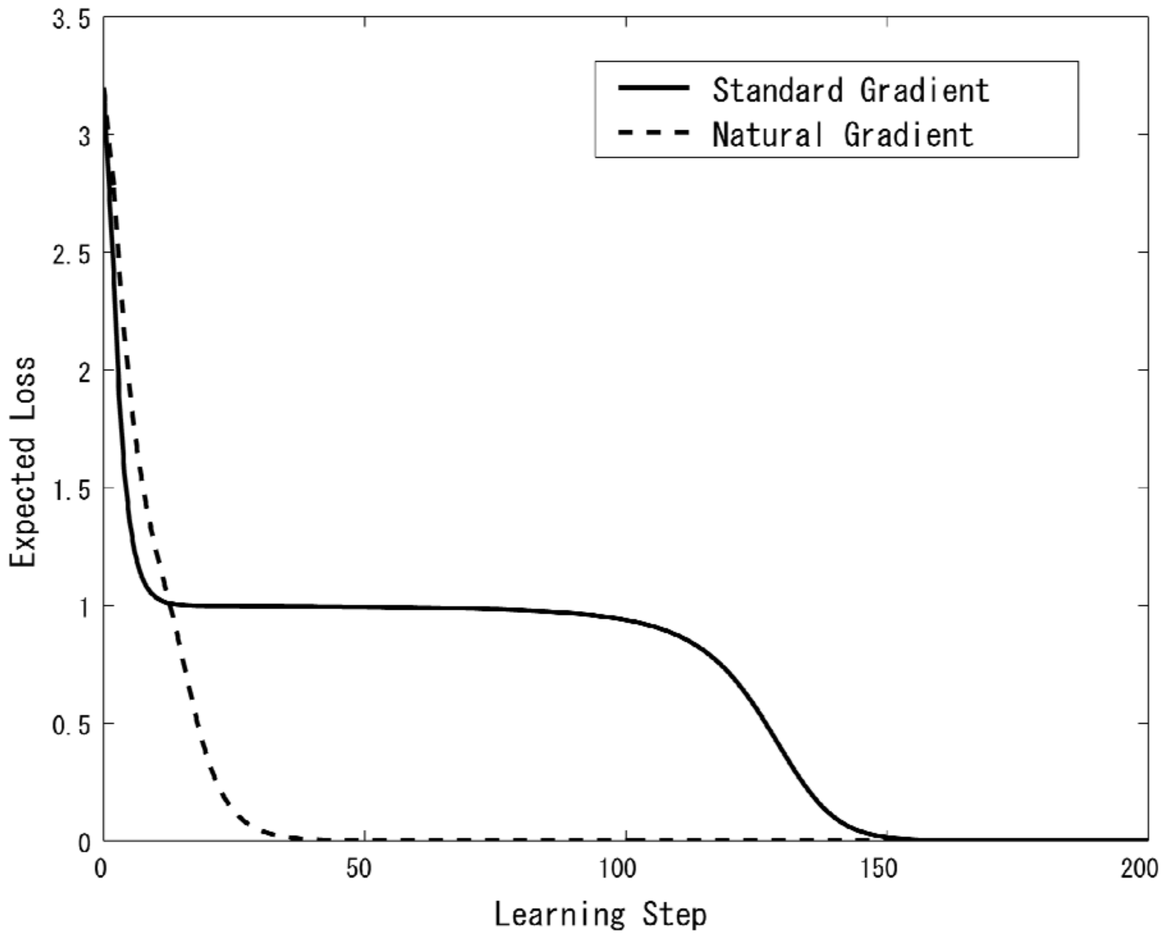


Dynamic vector fields: General case (|z|<1 part stable)



Dynamic vector fields: General case (|z|>1 part stable)

Adaptive Natural Gradient works well



Signal Processing

ICA : Independent Component Analysis

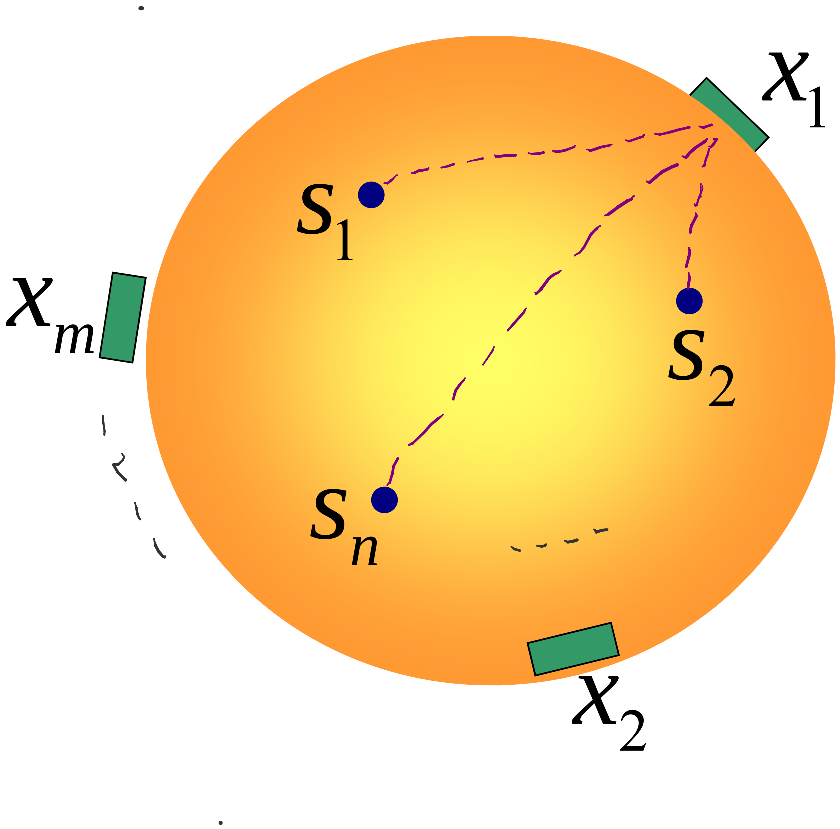
$$\boldsymbol{x}_t = A\boldsymbol{s}_t \qquad \boldsymbol{x}_t \to \boldsymbol{s}_t$$

$$\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} A \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix}$$

sparse component analysis

positive matrix factorization

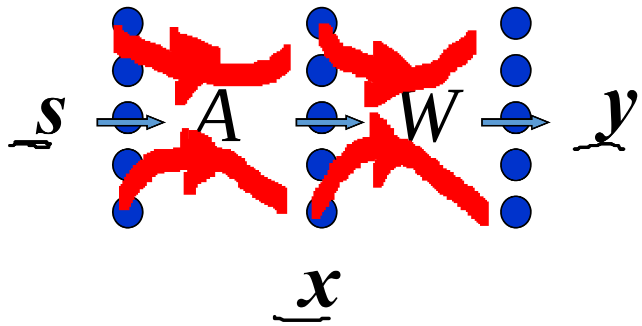
mixture and unmixture of independent signals



$$x_i = \sum_{j=1}^{n} A_{ij} s_j$$

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

Independent Component Analysis



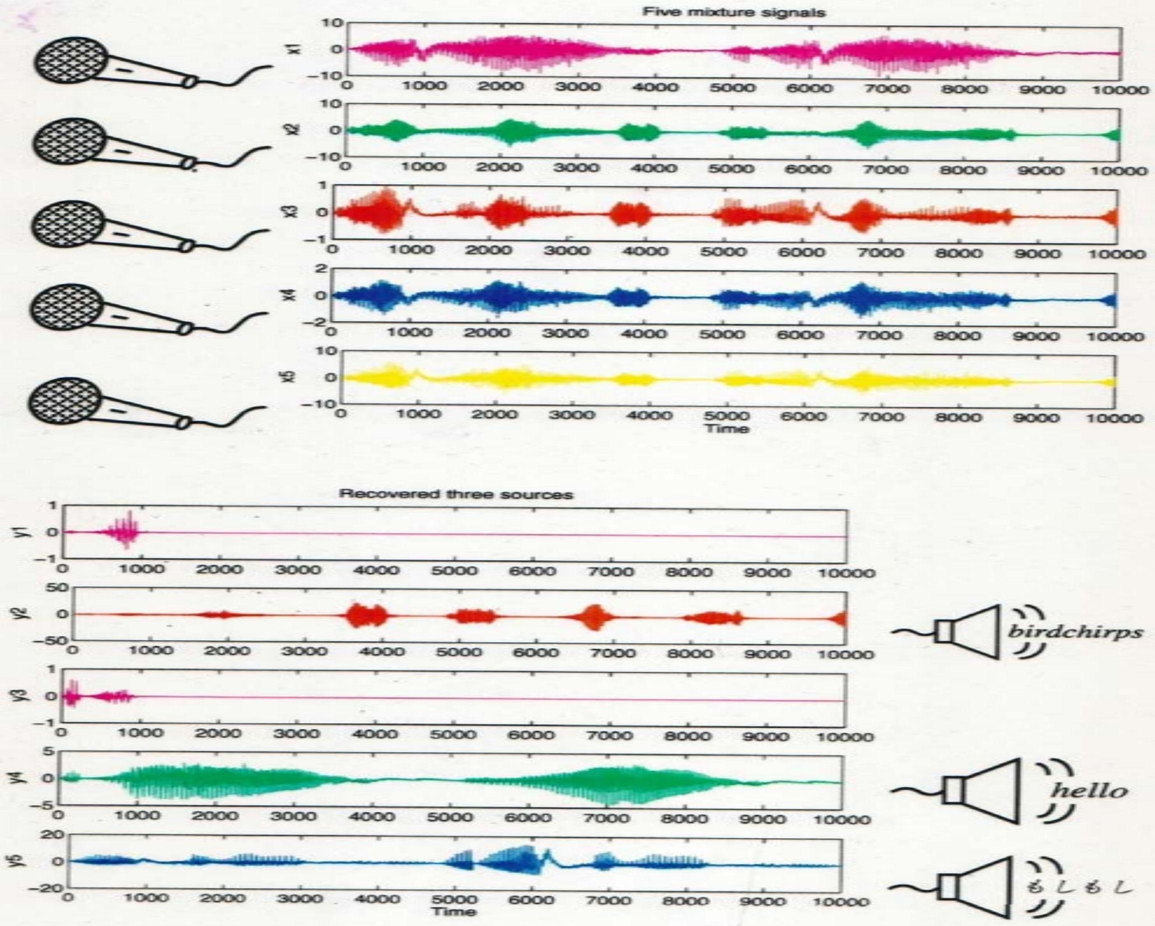
$$\boldsymbol{x} = A\boldsymbol{s} \qquad x_i = \sum A_{ij} s_j$$

$$\boldsymbol{y} = W\boldsymbol{x} \qquad W = A^{-1}$$

observations: x(1), x(2), …, x(t)

recover: s(1), s(2), …, s(t)

Cocktail party experiment



- 5 microphones (sensors) and only 3 speakers

Example of color image separation :

Five original images (but unknown to the neural net)

Five mixed images for separation

Final (stable states) of five separated images

Semiparametric Statistical Model

$$p(\mathbf{x};\mathbf{W},r) = |\mathbf{W}|\, r(\mathbf{W}\mathbf{x})$$

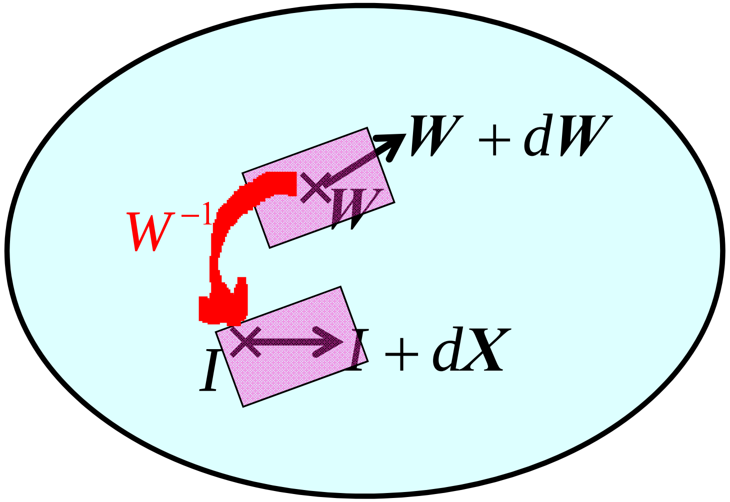
$\mathbf{W} = \mathbf{A}^{-1}$, $r(s)$: unknown $\qquad r = \prod r_i$

x(1), x(2), …, x(t)

Natural Gradient

$$\Delta \boldsymbol{W} = -\eta \frac{\partial l\left(\boldsymbol{y},\boldsymbol{W}\right)}{\partial \boldsymbol{W}} \boldsymbol{W}^T \boldsymbol{W}$$

Space of Matrices : Lie group



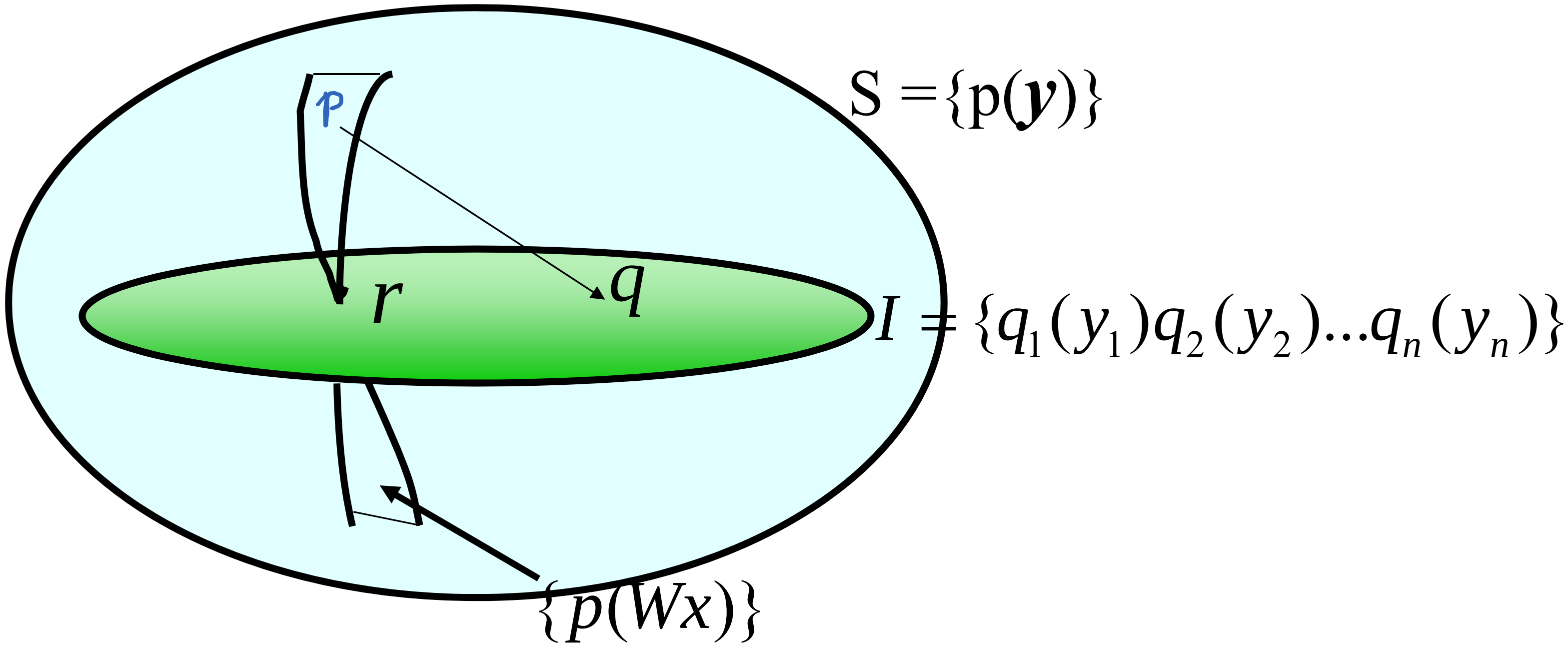
$$d\boldsymbol{X} = d\boldsymbol{W}\boldsymbol{W}^{-1}$$

$$\left| d\boldsymbol{W} \right|^2 = \mathrm{tr}\left(d\boldsymbol{X} d\boldsymbol{X}^T \right) = \mathrm{tr}\left(d\boldsymbol{W}\boldsymbol{W}^{-1}\boldsymbol{W}^{-T} d\boldsymbol{W}^T \right)$$

$$\widetilde{\nabla l} = \frac{\partial l}{\partial \boldsymbol{W}} \boldsymbol{W}^T \boldsymbol{W}$$

dX : **non-holonomic basis**

Information Geometry of ICA



natural gradient
estimating function
stability, efficiency

$$l(\mathbf{W}) = KL[\,p(\mathbf{y};\mathbf{W}) : q(\mathbf{y})]$$

$$r(\mathbf{y})$$

Estimating Functions

$$\Delta \boldsymbol{W} = -\eta \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{W})$$

$$E_{W,r}\left[\boldsymbol{F}\left(\boldsymbol{y}, \boldsymbol{W}'\right)\right] \begin{cases} = 0, & \boldsymbol{W}' = \boldsymbol{W} \\ \neq 0, & \boldsymbol{W}' \neq \boldsymbol{W} \end{cases}$$

estimating equation

$$\sum_t \boldsymbol{F}\left(\boldsymbol{y}_t, \boldsymbol{W}\right) = 0 \qquad \boldsymbol{y}_t = \boldsymbol{W}\boldsymbol{x}_t$$

learning

$$\Delta \boldsymbol{W}_t = -\eta \boldsymbol{F}\left(\boldsymbol{W}_t \boldsymbol{x}_t\right)$$

Admissible class

$$\boldsymbol{F}(\boldsymbol{y},\boldsymbol{W}) = \{\boldsymbol{I} - \boldsymbol{\varphi}(\boldsymbol{y})\boldsymbol{y}^T\}W$$

$$\tilde{\boldsymbol{F}} = \boldsymbol{R}(\boldsymbol{W})\boldsymbol{F}(\boldsymbol{y},\boldsymbol{W}) \qquad \{\alpha\varphi(y_i)y_j - \varphi(y_j)y_i\}W$$

$$\sum \boldsymbol{R}(\boldsymbol{W})\boldsymbol{F}(\boldsymbol{W}\boldsymbol{x}_t) = 0: \text{ estimating equation}$$

on-line learning : $\Delta\boldsymbol{W}_t = -\eta\boldsymbol{R}(\boldsymbol{W}_t)\boldsymbol{F}(\boldsymbol{W}_t\boldsymbol{x}_t)$

canonical estimating function: $\dfrac{\partial\tilde{F}}{\partial W} = I$

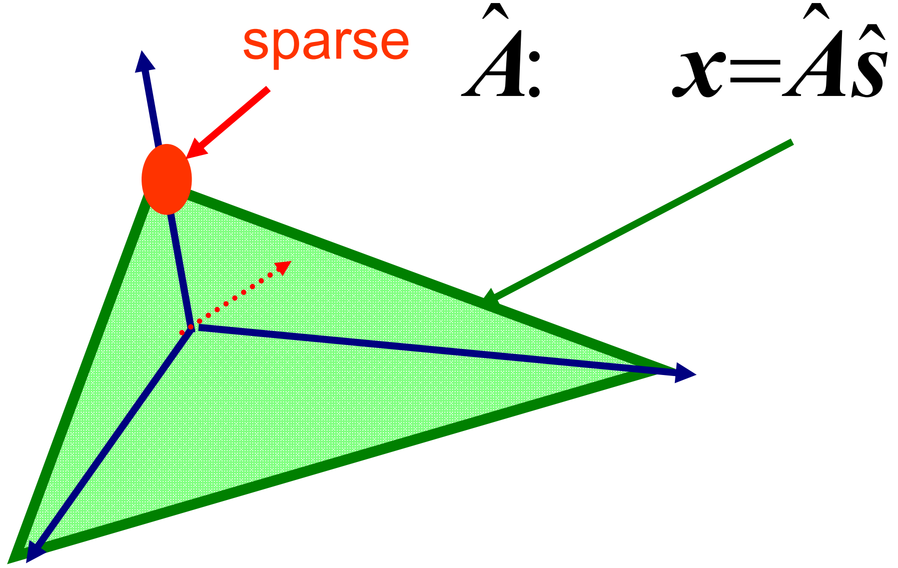
Basis Given: overcomplete case
Sparse Solution

$$\boldsymbol{x} = A\boldsymbol{s} = \sum s_i \boldsymbol{a}_i$$

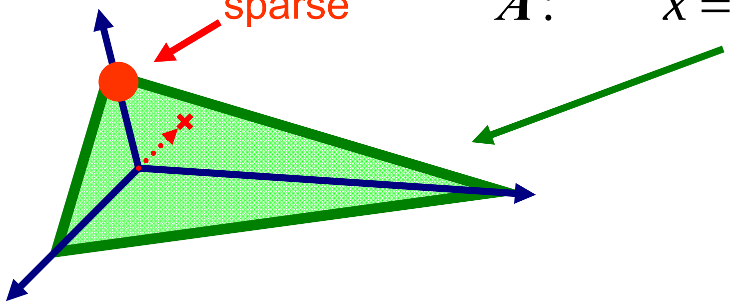
many solutions

many $s_i \to 0$

$$\boldsymbol{x}_t = \hat{A}\boldsymbol{s}_t$$



$$\hat{\boldsymbol{A}}: \qquad x = \hat{\boldsymbol{A}}\hat{\boldsymbol{s}}$$

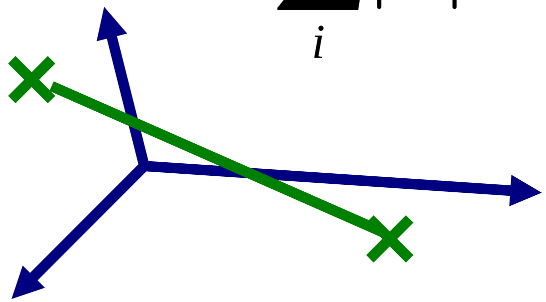


generalized inverse

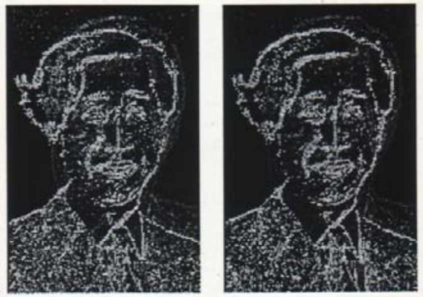
$$L_2\text{-norm:}\ \min \Sigma \left| \hat{\boldsymbol{s}}_{\boldsymbol{i}}^{2} \right|$$

sparse solution

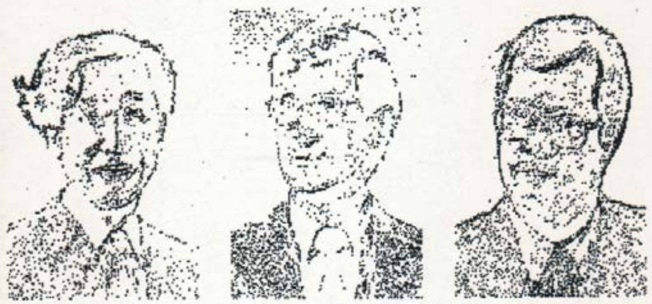
$$L_1\text{-norm:}\ \min \sum_i \left| \hat{\boldsymbol{s}}_i \right|$$



(a) Three binary edge images (reverse images are used in the experiment)



(b) Two edge image mixtures



(c) Reconstructed binary edge images (after reversion)

Fig. 5: Example of edge image image reconstruction: (a) the three binary edge images (reverse image copies are supplied for processing) , (b) their two mixtures, (c) the three extracted edge images (after reversion).

Minkovskian Gradient

Sparse Signal Analysis

$$\begin{cases} \min \psi(\boldsymbol{\beta}) & : \text{ convex function} \\ \text{constraint} & F(\boldsymbol{\beta}) \le c \end{cases}$$

typical case: $$\psi(\boldsymbol{\beta}) = \frac{1}{2}|\boldsymbol{y} - X\boldsymbol{\beta}|^2 = \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)^T G(\boldsymbol{\beta} - \boldsymbol{\beta}^*)$$

$$F(\boldsymbol{\beta}) = \frac{1}{p}\sum |\beta_i|^p \quad ; \qquad p = 2, \ \ p = 1, \ \ p = 1/2$$

Optimization under Sparsity Condition:

$$\boldsymbol{y} = X\boldsymbol{\beta} \quad (+\boldsymbol{n})$$

$$\boldsymbol{\beta}: \quad k\text{-sparse}$$

$$m > 2k \log n$$

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = m\left\{ \overbrace{\begin{bmatrix} & & \\ & X_{ij} & \\ & & \end{bmatrix}}^{} \right. \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}$$

$$\beta = \begin{pmatrix} 0 \\ 0 \\ \times \\ 0 \\ \times \\ 0 \\ \times \\ 0 \end{pmatrix} \begin{array}{l} \\ \\ \text{non-zero} \\ \\ k \\ \\ \\ \\ \end{array}$$

Linear regression

$$y_t = \sum_{i=1}^{k} x_i^t \beta_i + n_t, \qquad t = 1, 2, \ldots N$$

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{n} \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ \\ \mathrm{x}_N \end{bmatrix}$$

Overcomplete: N < k

under-determined → infinitely many solutions

Sparsity constraint solves the problem

Maximum likelihood estimator

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \sum_t \psi\left(y_t - \boldsymbol{\beta}\cdot\boldsymbol{x}_t\right), \quad \psi(u) = \frac{1}{2}u^2$$

$$\sum_t \psi'\left(y_t - \boldsymbol{\beta}\cdot\boldsymbol{x}_t\right)\boldsymbol{x}_t = 0$$

Euclidean case (Gaussian noise):

$$G = \nabla\nabla\psi = X^T X, \quad G\hat{\boldsymbol{\beta}} = X^T \boldsymbol{y}$$

$$\hat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T \boldsymbol{y} = \boldsymbol{X}^{\dagger}\boldsymbol{y}$$ Not sparse

Sparse Solution

$$\min \psi(\boldsymbol{\beta}) = \min D[\boldsymbol{\beta} : \boldsymbol{\beta}^*] = \psi(\boldsymbol{\beta}) - \psi(\boldsymbol{\beta}^*) - \nabla\psi(\boldsymbol{\beta}^*)(\boldsymbol{\beta} - \boldsymbol{\beta}^*)$$

$$\nabla\psi(\boldsymbol{\beta}^*) = 0$$

penalty $F_p(\boldsymbol{\beta}) = \sum |\boldsymbol{\beta}_i|^p = c$

$F_0(\boldsymbol{\beta}) = \#1[\beta_i \neq 0]$: **sparsest solution**

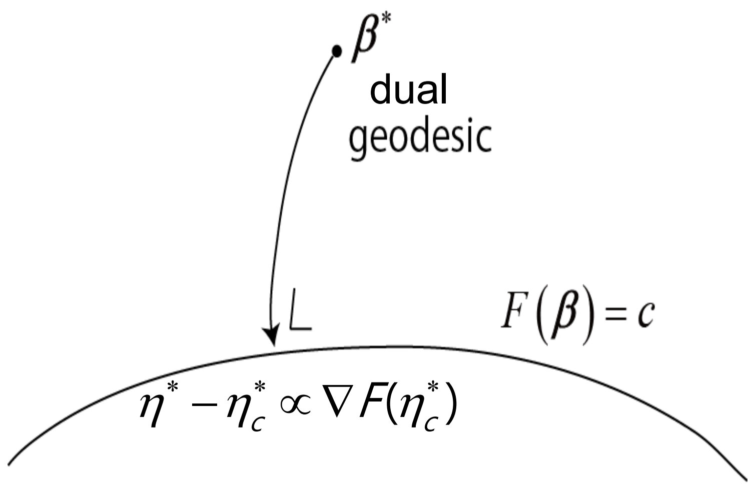
$F_1(\boldsymbol{\beta}) = \sum |\beta_i| : L_1$ **solution**

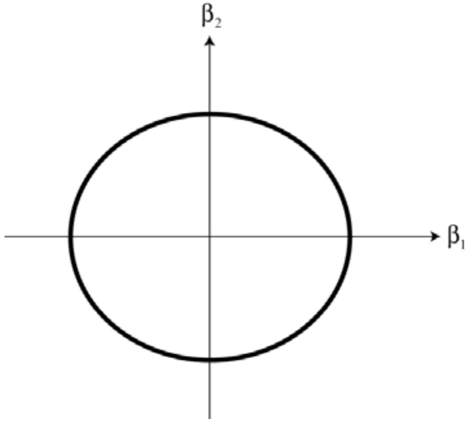
$F_p(\boldsymbol{\beta})$: $0 \leq p \leq 1$ Sparse solution: overcomplete case

$F_2(\boldsymbol{\beta}) = \sum |\beta_i|^2$: **generalized inverse solution**

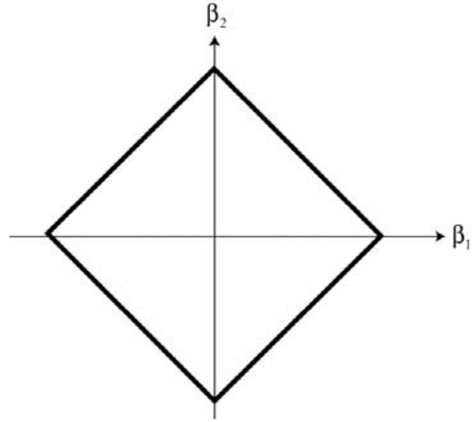
orthogonal projection, dual projection

$\min \psi(\boldsymbol{\beta}) = \mathbf{D}\left[\boldsymbol{\beta} : \boldsymbol{\beta}^*\right], \quad \mathbf{F}\left(\boldsymbol{\beta}\right) = \mathbf{c}$: **dual geodesic projection**

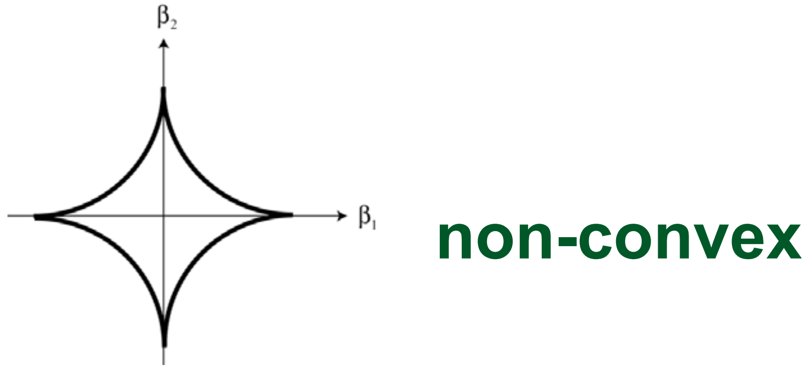




a) $R_c : n = 2, \;\; p > 1$



b) $R_c : n = 2, \; p = 1$



c) $R_c : n = 2, \; p < 1$

Fig. 1

L1-constrained optimization

P_c Problem

$\min \psi(\boldsymbol{\beta})$ under $F(\boldsymbol{\beta}) \le c$

solution $\boldsymbol{\beta}^*(c)$: $c = 0 \to \infty$

$\boldsymbol{\beta}_c^* = 0 \to \boldsymbol{\beta}^*$

LASSO

P_λ Problem

$\min \; \psi(\boldsymbol{\beta}) + \lambda F(\boldsymbol{\beta})$

LARS

solution $\boldsymbol{\beta}^*(\lambda)$ $\quad \lambda = \infty \to 0$

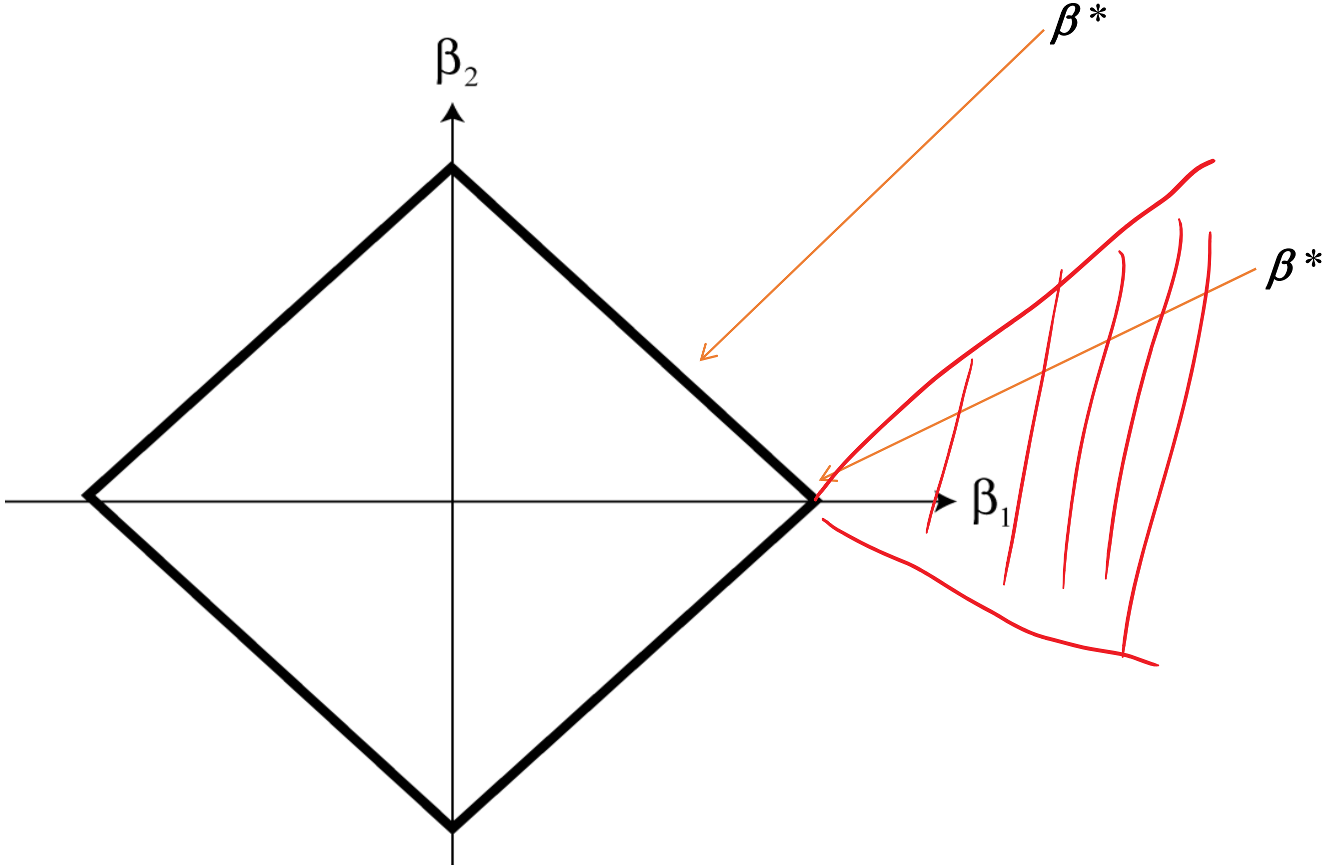
$\boldsymbol{\beta}_\lambda^* = 0 \to \boldsymbol{\beta}^*$

solutions $\boldsymbol{\beta}_c^*$ and $\boldsymbol{\beta}_\lambda^*$: coincide, $\lambda = \lambda(c)$, $p \ge 1$

$p < 1$: $\lambda = \lambda(c)$ multiple, noncontinuous

stability different

Projection from β^* to F = c (information geometry)



$$\eta_c^* \propto \nabla F(\eta_c^*)$$

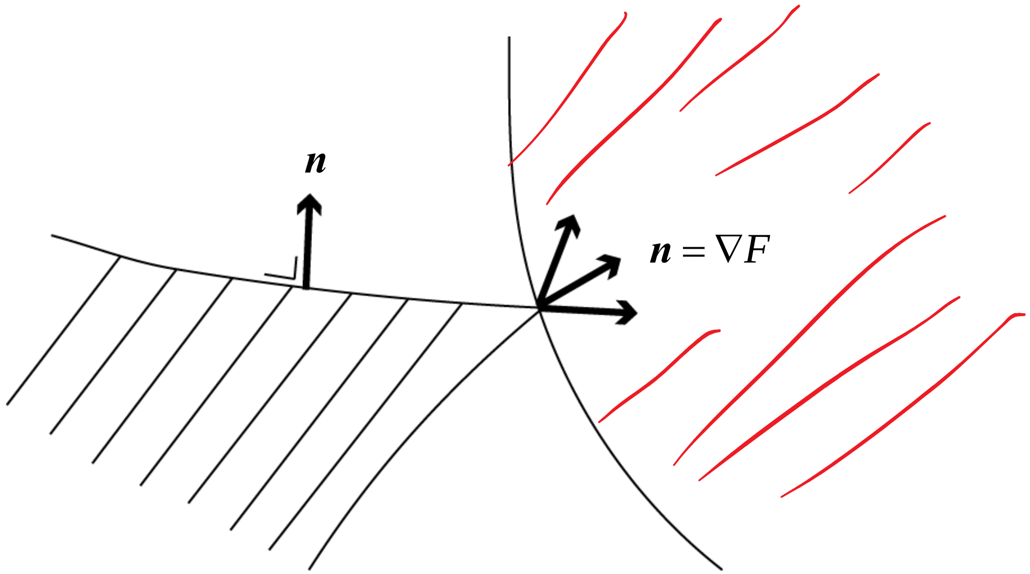


Fig. 5 subgradient

LASSO path and LARS path (stagewise solution)

$$\min \ \psi(\boldsymbol{\beta}) \ : \ F(\boldsymbol{\beta}) = c$$

$$\min \ \psi(\boldsymbol{\beta}) + \lambda F(\boldsymbol{\beta})$$

$\boldsymbol{\beta}^*(c), \boldsymbol{\beta}^*(\lambda)$ **c $\Leftrightarrow$ λ correspondence**

Active set and gradient

$$A(\boldsymbol{\beta}) = \{ i \mid \beta_i \neq 0 \}$$

$$\nabla F_p(\boldsymbol{\beta}) = \begin{cases} \operatorname{sgn}(\beta_i) \left| \beta_i \right|^{-(1-p)}, & i \in A \\ (-\infty, \infty), & i \notin A \\ [-1, 1] & \end{cases}$$

Solution path

$$\nabla_A \varphi\left(\boldsymbol{\beta}_c^*\right) + \lambda_c \nabla_A F\left(\boldsymbol{\beta}_c^*\right) = 0, \quad \boldsymbol{\beta}_c^*$$

$$\left\{\nabla_A \nabla_A \varphi\left(\boldsymbol{\beta}_c^*\right) + \lambda_c \nabla_A \nabla_A F\left(\boldsymbol{\beta}_c^*\right)\right\} \cdot \dot{\boldsymbol{\beta}}_c = -\dot{\lambda}_c \nabla_A F\left(\boldsymbol{\beta}_c\right)$$

$$\dot{\boldsymbol{\beta}}_c = -\dot{\lambda}_c K^{-1} \nabla_A F\left(\boldsymbol{\beta}_c^*\right) \;\; ; \;\; \dot{\boldsymbol{\beta}}_c = \frac{d}{dc} \boldsymbol{\beta}_c$$

$$K = G\left(\boldsymbol{\beta}_c^*\right) + \lambda_c \nabla \nabla F\left(\boldsymbol{\beta}_c^*\right)$$

$$\left(\nabla \nabla F_1 = 0; \quad \nabla F_1 = (\operatorname{sgn} \beta_i): \quad L_1\right)$$

Solution path in the subspace of the active set

$$\nabla_A \varphi\left(\boldsymbol{\beta}_\lambda^*\right) + \lambda \nabla_A F\left(\boldsymbol{\beta}_\lambda^*\right) = 0 \qquad \nabla_A \text{ : active direction}$$

$$\dot{\boldsymbol{\beta}}_\lambda^* = -K_A^{-1} \nabla_A F\left(\boldsymbol{\beta}_\lambda^*\right)$$

turning point $\mathbf{A} \to \mathbf{A}'$

Gradient Descent Method

$$\min \mathrm{L}(\beta + \mathrm{a}): \quad g_{ij} a^i a^j = \varepsilon^2$$

$$\nabla L = \{\frac{\partial}{\partial \beta_i} L(\beta)\}: \quad \text{covariant}$$

$$\tilde{\nabla} L = \{\sum g^{ji} \frac{\partial}{\partial \beta_i} L(\beta)\}: \quad \text{contravariant}$$

$$\beta_{t+1} = \beta_t - c\widetilde{\nabla} L(\beta_t)$$

Steepest direction of L

$$G(\boldsymbol{a}) = \sum g_{ij}(\boldsymbol{\beta}) a_i a_j$$ **Riemannian metric**

$$G(t\boldsymbol{a}) = |t|^p \, G(\boldsymbol{a}) > 0$$ **Minkovskian metric**

$$G(\boldsymbol{a}) = \sum |a_i|^p. \qquad p > 1$$ **Lp-norm**

$$\frac{\delta}{\delta \boldsymbol{a}} \left\{ \nabla L \cdot \boldsymbol{a} - \lambda G(\boldsymbol{a}) \right\} = 0$$

$$\delta G(\boldsymbol{a}) = p(\text{sign } a_i)|a_i|^{p-1}$$

$$f_i = \frac{\partial}{\partial \beta_i} F(\boldsymbol{\beta})$$

$$a_i = c(\text{sgn } f_i)|f_i|^{\frac{1}{p-1}}$$

$$G(\boldsymbol{a}) = \sum g_{ij} a_i a_j, \qquad \tilde{\nabla} F = G^{-1} \nabla f$$

Natural gradient

$$\tilde{\nabla} F = \nabla f$$

Euclidean case

$$\tilde{\nabla} F = c\left(\operatorname{sgn} f_i\right)\left| f_i \right|^{\frac{1}{p-1}}$$

$$\tilde{\nabla} F = c\left(\operatorname{sgn} f_{i^*}\right)\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\alpha \to 1$$

$$i^* = \arg\max |f_i|$$

$$\max |f_i| = |f_{i^*}| = |f_{j^*}|$$

$$\left(\tilde{\nabla} F\right)_i = \begin{cases} 1, & \text{for } i = i^* \text{ and } j^*, \\ 0 & \text{otherwise.} \end{cases}$$

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \eta \tilde{\nabla} F$$

LASSO

Try for various p, p->1
Try for various noise function
LASSO and flat geometry

Extended LARS (p = 1) and Minkovskian gradient

norm $\|\boldsymbol{a}\|_p = \sum |a_i|^p$

$\max \psi(\boldsymbol{\beta} + \varepsilon \boldsymbol{a}) \qquad$ under $\|\boldsymbol{a}\|_p = 1$

$\psi(\boldsymbol{\beta} + \varepsilon \boldsymbol{a}) - \lambda \|\boldsymbol{a}\|_p$

$p = 1^+$

$$\nabla_1 \psi(\boldsymbol{\beta}) = \mathbf{1}_A \begin{cases} \operatorname{sgn} \eta_i, & |\eta_i| = \max\{|\eta_1|, \cdots, |\eta_N|\} \\ 0, & \text{otherwise} \end{cases}$$

$\boldsymbol{\eta} = \nabla \psi(\boldsymbol{\beta})$

$$i^* = \arg\max |f_i|$$

$$\max |f_i| = |f_{i^*}| = |f_{j^*}|$$

$$\left(\tilde{\nabla} F\right)_i = \begin{cases} 1, & \text{for } i = i^* \text{ and } j^*, \\ 0 & \text{otherwise.} \end{cases}$$

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \eta \tilde{\nabla} F$$

LARS

L1/2 constraint: non-convex optimization

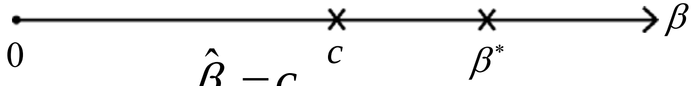
λ-trajectory and σ-trajectory

Ex. **1-dim**

$$\varphi(\beta) = \frac{1}{2}(\beta - \beta^*)^2$$

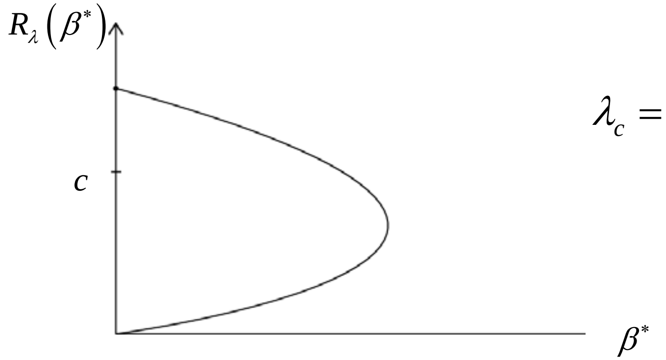
$$f_\lambda(\beta) = \phi + \lambda F = \frac{1}{2}(\beta - 2)^2 + 2\lambda\sqrt{\beta}$$

$$P_c : \min\left(\beta - \beta^*\right)^2, \quad |\beta| \le c$$



$$\hat{\beta}_c = c$$

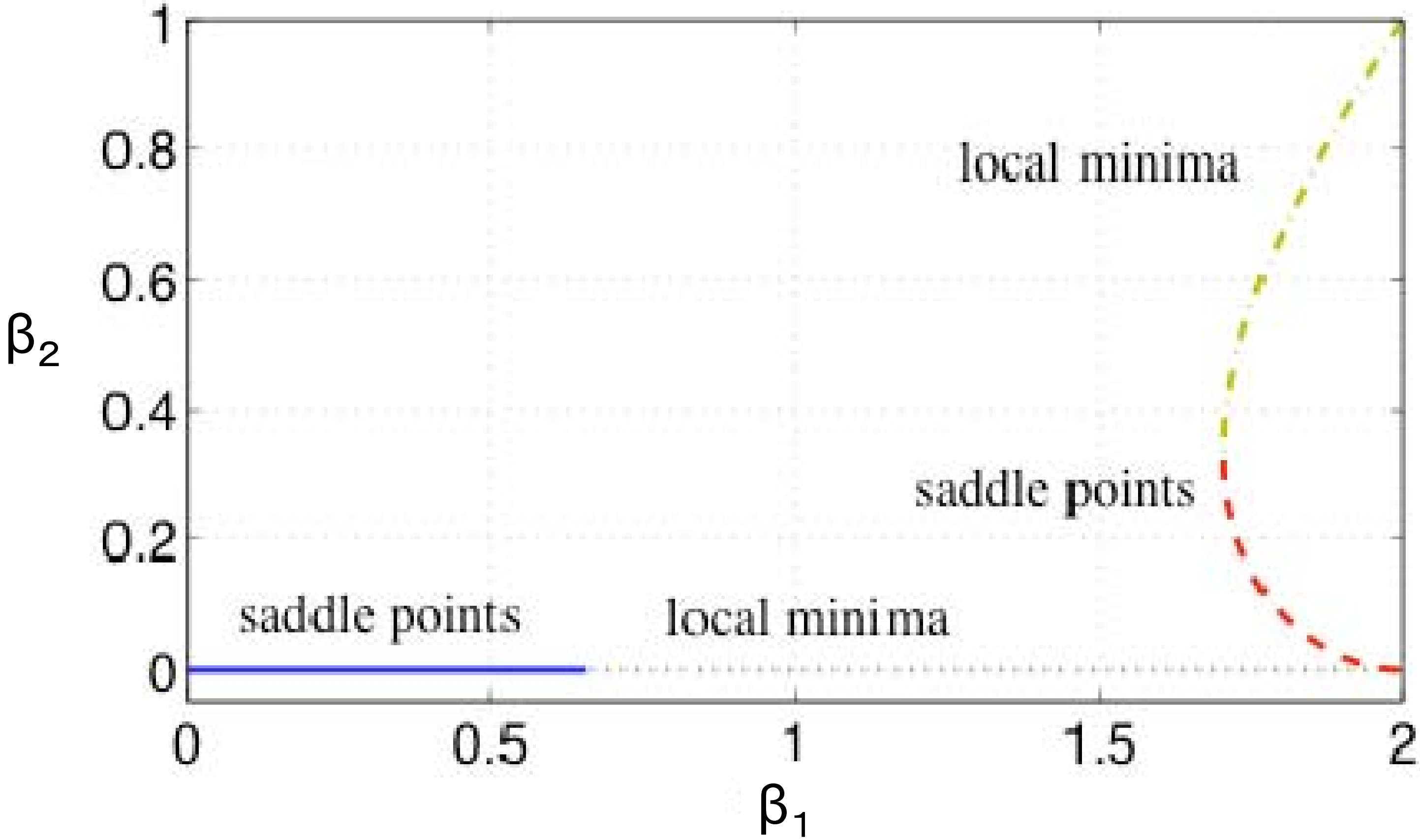
$P_\lambda : \nabla f_\lambda = 0$ $\quad$ $\beta - \beta^* + \dfrac{\lambda}{\sqrt{\beta}} = 0$ $\quad$ $\hat{\beta} = R_\lambda\left(\beta^*\right)$: Xu Zongben's operator



$$\lambda_c = \sqrt{c}\left(\beta^* - c\right)$$

λ

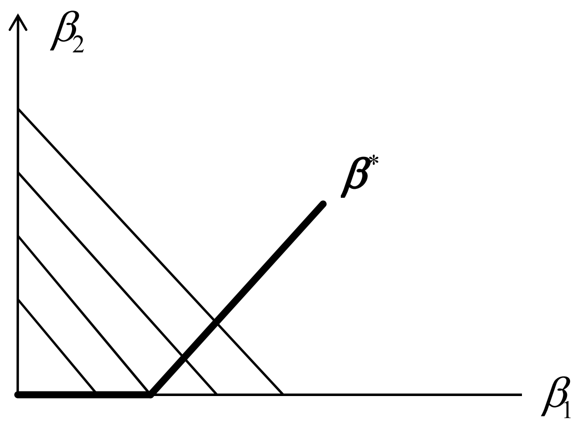
An Example of the greedy path



LASSO and LARS :

$p > 1$: $\boldsymbol{\beta}_c^*$ is non-sparse

$p = 1$: sparse

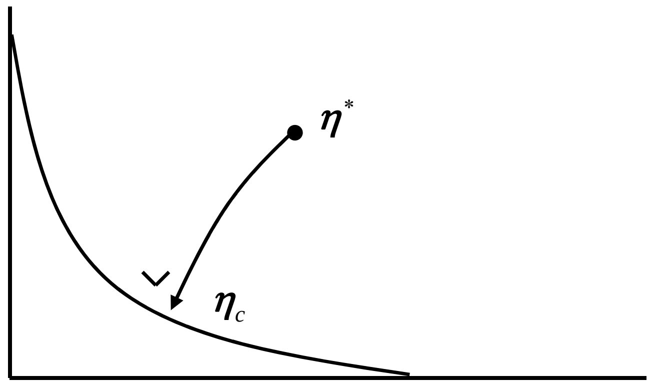


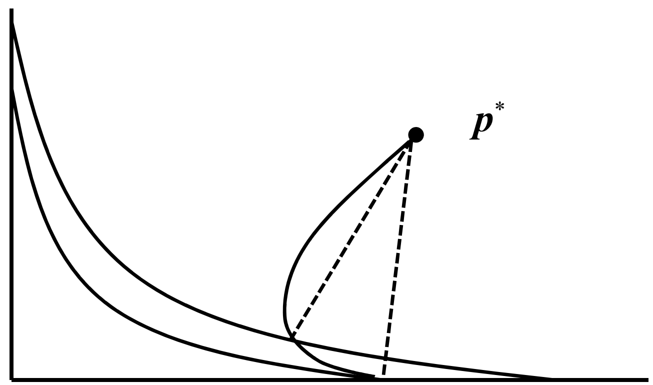
Minkovskian gradient

$$D\left[\boldsymbol{\beta}_c : \boldsymbol{\beta}^*\right] - \lambda F\left(\boldsymbol{\beta}\right)$$

$$\nabla\varphi\left(\boldsymbol{\beta}_c\right) - \nabla\varphi\left(\boldsymbol{\beta}^*\right) - \lambda_c \nabla F\left(\boldsymbol{\beta}_c\right) = 0$$

$$\boldsymbol{\eta}_c - \boldsymbol{\eta}^* = \lambda_c \nabla F\left(\boldsymbol{\beta}_c\right)$$





$$\nabla\nabla\varphi\left(\boldsymbol{\beta}_c\right)\cdot\dot{\boldsymbol{\beta}}_c - \lambda_c\nabla\nabla F\left(\boldsymbol{\beta}_c\right)\cdot\dot{\boldsymbol{\beta}}_c = \dot{\lambda}_c\nabla F\left(\boldsymbol{\beta}_c\right)$$

G H

$$\dot{\boldsymbol{\beta}}_c = \left(G - \lambda_c H\right)^{-1}\nabla F\left(\boldsymbol{\beta}_c\right)$$

Solution Path : $\lambda \leftrightarrow c$

not continuous, not-monotone jump

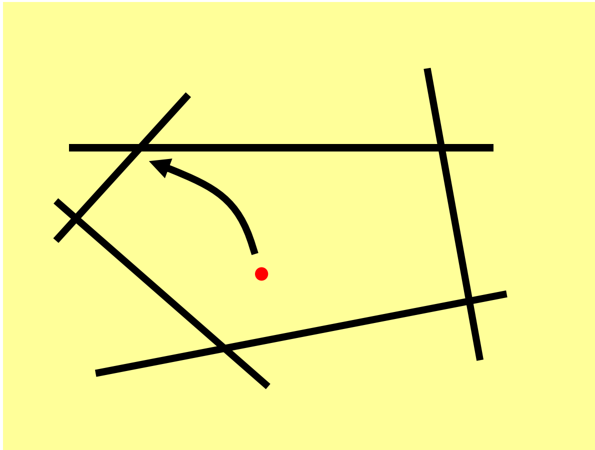
$$\beta_{\lambda} \Leftrightarrow \beta_{c}$$

Linear Programming

$$\sum A_{ij} x_j \geq b_i$$

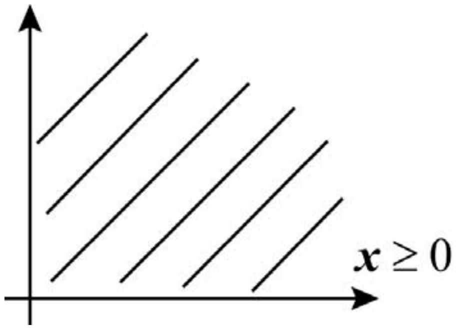
$$\max \sum c_i x_i$$

$$\psi(\boldsymbol{x}) = \sum_i \log\left(\sum A_{ij} x_j - b_i\right)$$



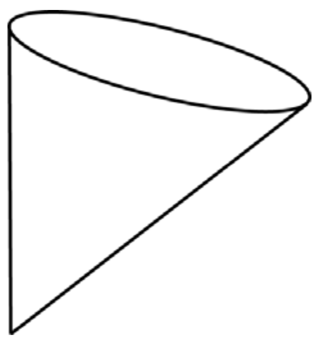
Convex Cone Programming

P : positive semi-definite matrix



convex potential function

dual geodesic approach



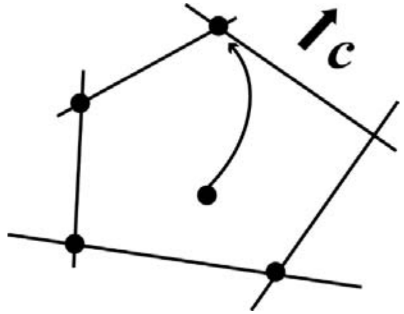
$$A\boldsymbol{x} = \boldsymbol{b}, \quad \min \boldsymbol{c} \cdot \boldsymbol{x}$$

Support vector machine

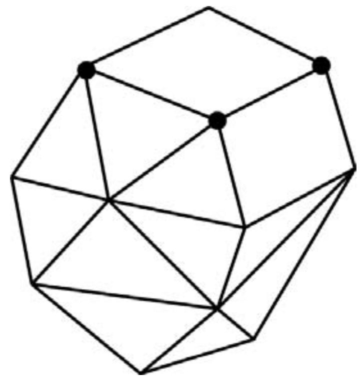
Convex Programming
— Inner Method

$$LP: A\boldsymbol{x} \geq \boldsymbol{b}$$

$$\min \quad \boldsymbol{c} \cdot \boldsymbol{x}$$



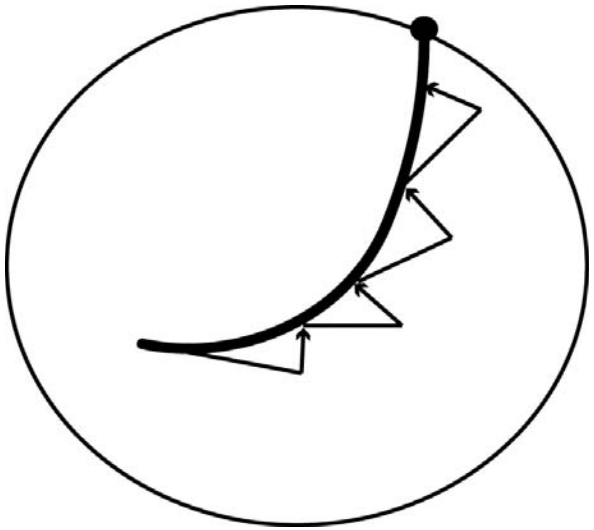
Barrier function $$\psi(\boldsymbol{x}) = \sum \log\left(\sum A_{ij} x_j - b_i\right)$$



$$\boldsymbol{\eta} = \partial_i \psi(\boldsymbol{x})$$

Simplex method ; inner method

Polynomial-Time Algorithm



curvature : step-size

$$\left|\boldsymbol{H}^{(m)}\right|^2$$

$$\min : t\boldsymbol{c}\cdot\boldsymbol{x}+\psi(\boldsymbol{x}) \qquad \boldsymbol{x}=\boldsymbol{\delta}(t) \qquad \nabla^{*}-\text{geodesic}$$

Integration of evidences:

$$x_1, x_2, \ldots x_m$$

arithmetic mean
geometric mean
harmonic mean
α-mean

Generalized mean: f-mean

f(u): monotone; f-representation of u

$$m_f(a,b) = f^{-1}\left\{\frac{f(a)+f(b)}{2}\right\}$$

scale free $\quad m_f(ca,cb) = c\,m_f(a,b)$

α-representation $\quad f_\alpha(u) = \begin{cases} u^{\frac{1-\alpha}{2}}, & \alpha \neq 1 \\ \log u, & \alpha = 1 \end{cases}$

α-mean : $m_\alpha(p_1(s), p_2(s))$

$$\alpha = 1 \quad : \quad \sqrt{ab}$$

$$\alpha = -1 : \quad \frac{a+b}{2}$$

$$\alpha = 0 \quad : \quad (\sqrt{a} + \sqrt{b})^2 = \frac{a+b}{4} + \frac{1}{2}\sqrt{ab}$$

$$\alpha = \infty \qquad m_\alpha = \min(a, b)$$

$$\alpha = -\infty \qquad m_\alpha = \max(a, b)$$

Various Means

$$\frac{a+b}{2} \quad : \quad \sqrt{ab} \quad : \quad \frac{2}{\frac{1}{a}+\frac{1}{b}}$$

arithmetic **geometric** **harmonic**

Any other mean?

α — Family of Distributions

$\{p_1(s), \cdots, p_k(s)\}$

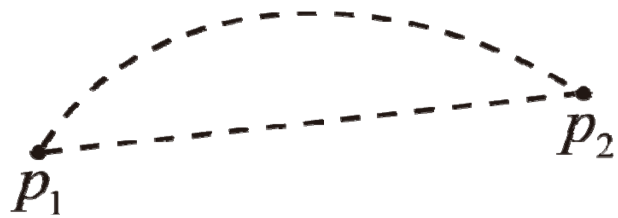
$$p(x;\theta) = f_\alpha^{-1}\{\sum \theta_i f_\alpha(p_i(x))\}$$

mixture family :

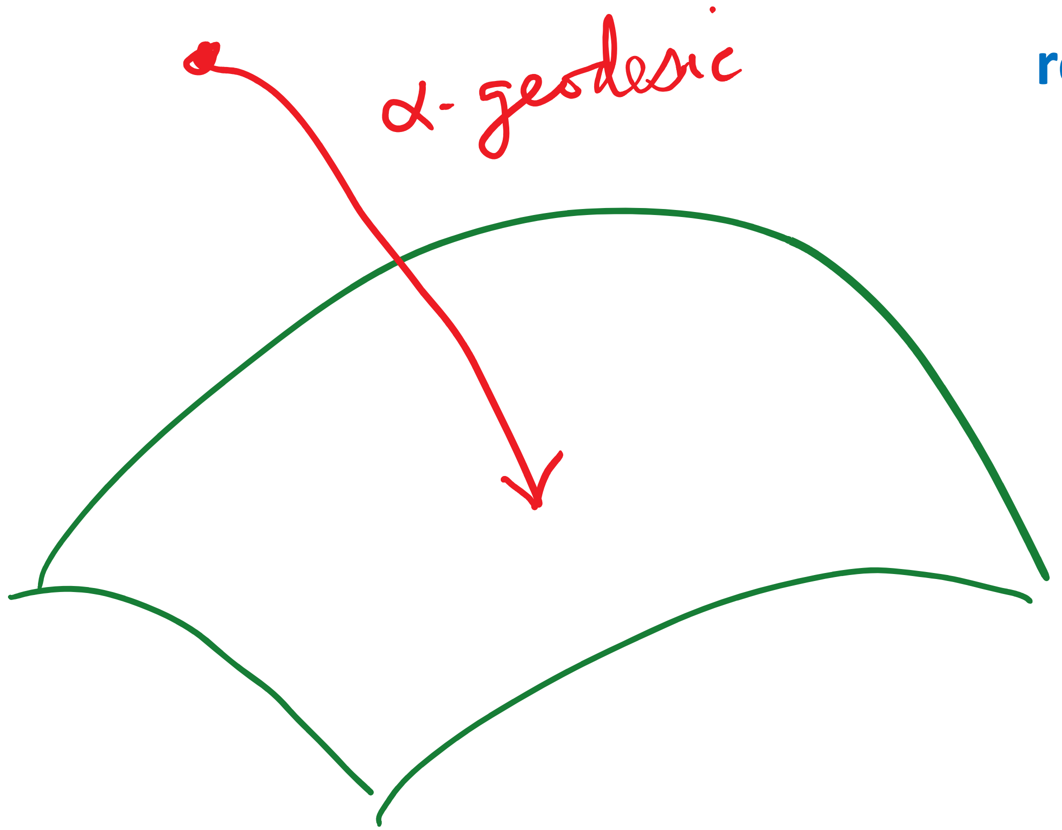
$$p_{mix}(s) = \sum_{i=1}^{k} t_i p_i(s), \qquad \sum t_i = 1$$

exponential family :

$$\log p_{\exp}(s) = \sum t_i \log p_i(s) - \psi$$



α-geodesic projection



robust estimator