Kernel Methods: From Image Analysis to Deep Learning

Yunmei Chen

University of Florida

May, 2017

Outline

- Kernel method for deformable multi-modal image registration
- Kernel method for non-parametric segmentation
- Kernel method in deep learning

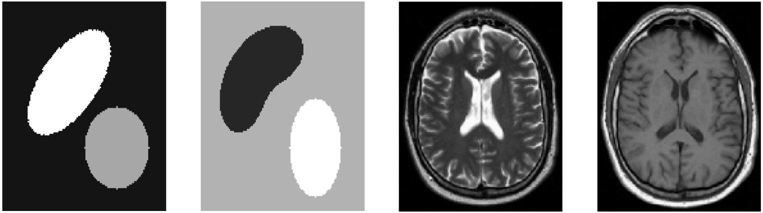
I. Kernel Method For Deformable Multi-Modal Image Registration

Uni-model image registration

- Image registration finds a deformation field u that aligns a pair of images S, T:
$$S(x+u(x)) \approx T(x).$$
- Image registration allows an accurate fusion of complementary information. It has been widely used to assist diagnosis and treatment in health care.
- General variational framework for image registration:
$$min_u \lambda Reg(u) + dis(S(x+u(x)), T(x)).$$
- ex. for $dis(S(x+u(x)), T(x))$:
$$CC(S(x+u(x)), T(x)) \; or \; \|S(x+u(x)) - T(x)\|^2$$

Multi-modal Image Registration

- There is no direct comparison for intensities of multi-modal images.
- What is a good similarity measure?
- Can we compare the intensities of transformed multi-modal images and how to find such a transform?



Is any good similarity measure for multi-modal images?

Information theoretical approach: Maximizing Mutual Information (MI)

- Model:

$$\begin{aligned} \min E(u(x)) &= \lambda \int_\Omega |\nabla u(x)|^2 dx + MI(S(x+u(x)), T(x)) \\ MI(X,Y) &= \int_R \int_R p_{X,Y}(i,j) \log \frac{p_{X,Y}(i,j)}{p_X(i) p_Y(j)} didj \end{aligned}$$

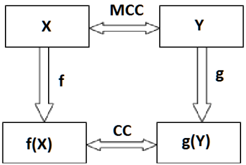
- Joint PDF Estimator
  $p_{X,Y}(i,j) = \frac{1}{\Omega} \int_\Omega K_h(S(x+h(x)) - i, T(x) - j) dx.$
  K_h: smooth, positive, rapidly decreasing to zero outside the window O_h, $\int K_h = 1$.
- Limitation: The need of the estimation of joint pdf makes the algorithm complex and sensitive to image quantization.

Can we find non-linear transforms/maps, s.t. the transformed images are linearly related?

Renyi's dep. meas.: maximum correlation coefficient $MCC(X, Y)$

$$MCC(X, Y) = \sup_{f,g \in V} CC(f(X), g(Y))$$

V : Borel measurable functions with finite positive variance.



- X and Y nonlinearly related, but $f(X)$ and $g(Y)$ linearly related

Computation for Renyi's Dependence Measure: MCC

$$MCC(X, Y) = \sup_{f,g \in V} CC(f(X), g(Y)),$$

$$CC(f(X), g(Y)) = \frac{Cov(f(X), g(Y))}{\sqrt{Var(f(X))}\sqrt{Var(g(Y))}},$$

$$Cov(f(X), g(Y)) = E[(f(X) - E(f(X)))(g(Y) - E(g(Y))].$$

- Advantage: to compute $MCC(X, Y)$ we do not deal with the joint PDF itself, but instead, observed samples drawn independently according to it.
- Difficulty: the space V is too large, finding supreme in V is difficult.
- The difficulty can be overcome by using Reproducing Kernel Hilbert Space (RKHS).

What is RKHS

- RKHS is a Hilbert space of func.s, in which the point eval.s

$$L_x : L_x(f) = f(x), \ \ \forall x \in dom(f)$$

  are bounded linear functionals i.e. $L_x \in H'$.
- A Hilbert space H is RKHS $\Leftrightarrow$ it has a reproducing kernel.
  - H is a RKHS, $L_x \in H'$. By Riesz rep. thm. $\exists$ an unique $K_x \in H$, s.t.

$$f(x) = L_x(f) =< f, K_x > .$$

  - Let $K(x, \cdot) = K_x(\cdot)$, and $K(x, x') \triangleq < k_x, k_x' >$. Then

$$< K(x, x'), f(x') >=< k_x, f >= f(x),$$

$$< K(x, y), K(x', y) >= K_{x'}(x) = K(x, x').$$

  $K(x, x')$ is a sym., PD and rep. kernel for H.

What is RKHS (cont.)

- Moore-Aronszajn Thm.: for every sym., PD kernel $K(\cdot,\cdot)$, there is a unique Hilbert space H, for which K is a reproducing kernel.
  - Let
  $$H_0(\Omega) = \{f|f(x) = \sum_{i=1}^{n} \alpha_i K(x, y_i),\ \forall n,\ \forall y_i \in \Omega\}$$
  - Let $H(\Omega)$ be the completion of $H_0(\Omega)$ w.r.t. the inner product
  $$\langle K(x_i, y), K(x_j, y)\rangle = K(x_i, x_j).$$

  $H(\Omega)$ is the unique RKHS associated with the kernel K.
- Example: Let $K(x,y) = K_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma}} \exp\{-\frac{|x-y|^2}{2\sigma^2}\}$. Then,
$$f(x) = \sum_{i=1}^{m} \frac{\alpha_i}{\sqrt{2\pi\sigma}} \exp\{-\frac{|x-y_i|^2}{2\sigma^2}\},\quad \forall f \in H_0(\Omega).$$

Theorem

$$\sup_{f,g\in H_0(R)} CC(f(X),g(Y)) = \sup_{f,g\in V(R)} CC(f(X),g(Y))$$

Lemma

$H_0(R)$ is dense in $C_0(R)$

Lemma

$$\sup_{f,g\in C_0(R)} \mathbf{CC}(f(X),g(Y)) = \sup_{f,g\in V_B(R)} \mathbf{CC}(f(X),g(Y))$$

where $V_B(R)$ is the space of bounded measurable functions on R.

Lemma

$$\sup_{f,g\in V_B(R)} \mathbf{CC}(f(X),g(Y)) = \sup_{f,g\in V(R)} \mathbf{CC}(f(X),g(Y))$$

Proposed Model

$$\arg\min E(u(x), a_i, b_j) = \lambda \int_\Omega |\nabla u(x)|^2 dx + (1 - CC(M, N))^2$$

- $M(x) = f(S(x + u(x))) = \sum_i^m a_i K\sigma(S(x + u(x)), y_i)$
- $N(x) = g(T(x)) = \sum_j^n b_j K\sigma(T(x), z_j)$
- Model enforces $CC(M, N) \approx 1$ to maximize the statistical dependence of the deformed and target images.

Evolution Equations

$$\partial u/\partial t = 2\lambda \triangle u(x,t) + (1 - CC(M,N))F(x,t), \quad x \in \Omega, \quad t > 0,$$
$$\partial u/\partial n = 0, \quad x \in \partial\Omega, \ t > 0, \qquad u(x,t) = 0, \quad x \in \Omega,$$

$$\frac{da_i}{dt} = (1 - CC(M,N))\frac{cov(P_i,N)\cdot var(M) - cov(M,N)\cdot cov(M,P_i)}{[var(M)]^{\frac{3}{2}}\cdot[var(N)]^{\frac{1}{2}}},$$
$$\frac{db_i}{dt} = (1 - CC(M,N))\frac{cov(Q_i,M)\cdot var(N) - cov(M,N)\cdot cov(N,Q_i)}{[var(N)]^{\frac{3}{2}}\cdot[var(M)]^{\frac{1}{2}}},$$

where

$$\begin{aligned} F &= \frac{1}{|\Omega|}[(N - \bar{N})\cdot var(M) - (M - \bar{M})\cdot cov(M,N)] \\ &\cdot \quad [var(M)]^{-3/2}\cdot[var(N)]^{-1/2}\cdot\nabla M \end{aligned}$$

Local Version of the MCC Model

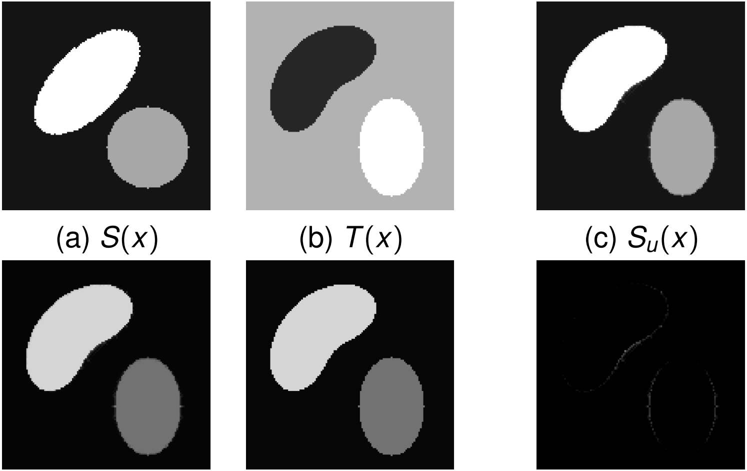
$$\min_{u(x),\alpha(x),\beta(x)} \lambda \int_\Omega |\nabla u(x)|^2 dx + |\Omega| \int_\Omega (1 - \frac{\nu_{12}(x)}{\sqrt{\nu_1(x)} \cdot \sqrt{\nu_2(x)}})^p dx,$$

$$\nu_{12}(x) = Cov_{O_x}(f(S(x+u(x)), g(T(x))), \;\; O_x : nbd \; of \; x,$$

$$\nu_1(x) = Var_{O_x}(f(S(x+u(x))), \;\; \nu_2(x) = Var_{O_x}(g(T(x)).$$

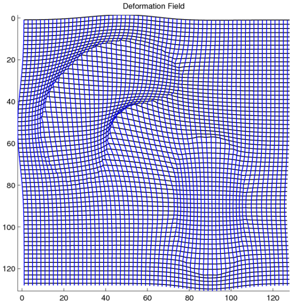
- Advantage:
  Local version provides the flexibility to cope with complex intensities of S and T. Since S and T take less values locally, m can be smaller in $f(S(x+u(x))) = \sum_i^m a_i K_\sigma(S(x+u(x)), y_i)$, $g(T(x)) = \sum_j^m b_j K_\sigma(T(x), z_j)$.

Accuracy of the MCC method for synthetic images

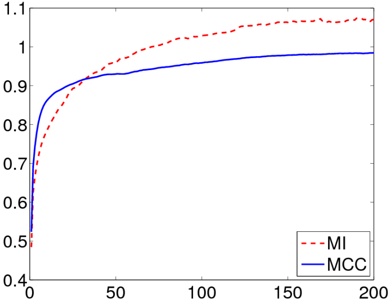


(a) $S(x)$ (b) $T(x)$ (c) $S_u(x)$

(e) $f(S_u)(x)$ (f) $g(T)(x)$ (g) $|f(S_u)(x) - g(T)(x)|$

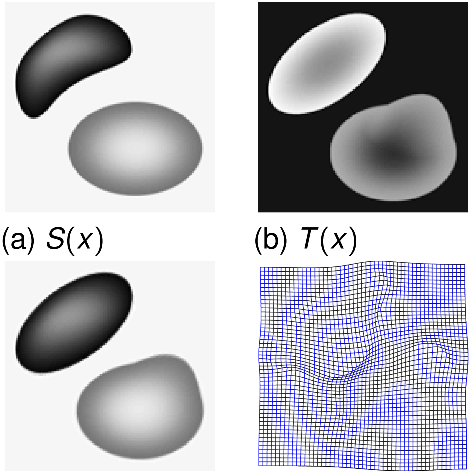


(a) $u(x)$



(b) MCC and MI

Robustness to inhomogeneity & bias of intensities



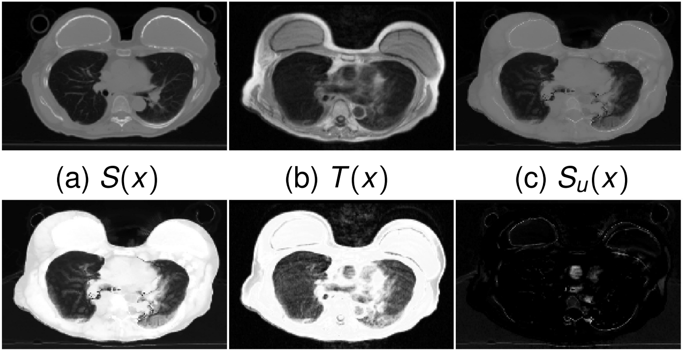
(a) $S(x)$

(b) $T(x)$

(c) $S_u(x)$

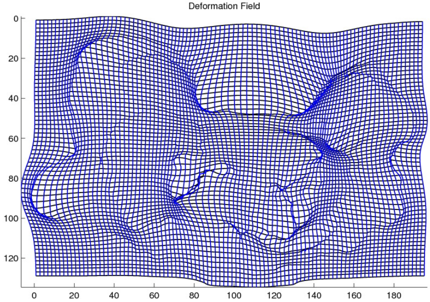
(d) $u(x)$

CT and MRI lung image registration

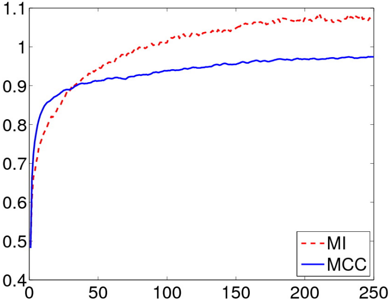


(a) $S(x)$ (b) $T(x)$ (c) $S_u(x)$

(e) $f(S_u)(x)$ (f) $g(T)(x)$ (g) $|f(S_u)(x) - g(T)(x)|$

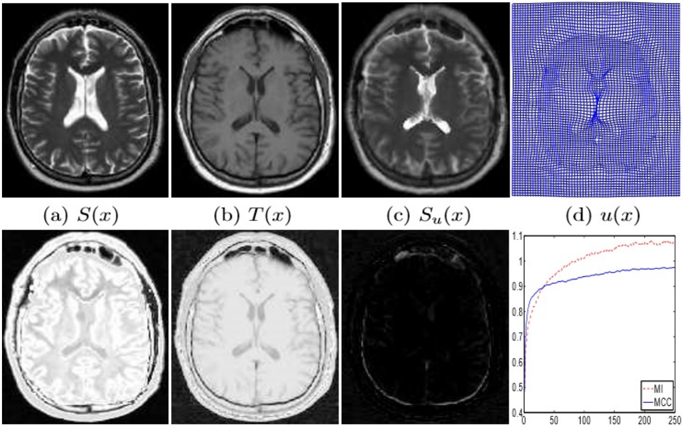


(a) $u(x)$



(b) MCC and MI

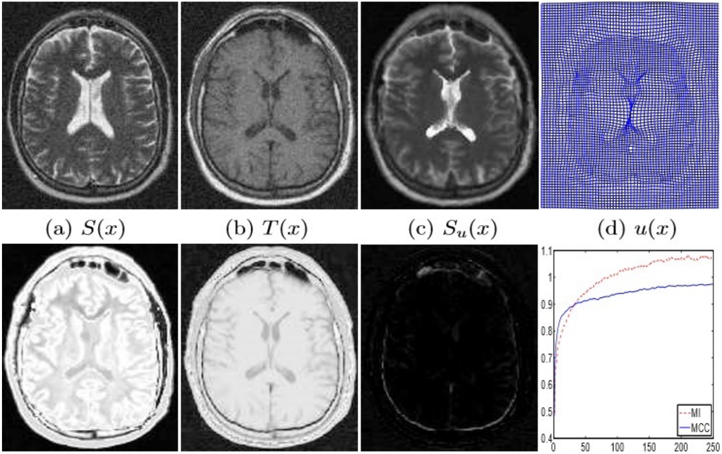
T1 and T2 brain image registration - Local MCC



(a) $S(x)$ (b) $T(x)$ (c) $S_u(x)$ (d) $u(x)$

(e) $S_u^{f^*}(x)$ (f) $T^{g^*}(x)$ (g) $|S_u^{f^*}(x) - T^{g*}(x)|$ (h) MCC and MI

T1 and T2 noisy brain image registration (N(0,0.1))



(a) $S(x)$ (b) $T(x)$ (c) $S_u(x)$ (d) $u(x)$

(e) $S_u^{f^*}(x)$ (f) $T^{g^*}(x)$ (g) $|S_u^{f^*}(x) - T^{g*}(x)|$ (h) MCC and MI

Table: Sensitivity on the kernel size σ.

model using MI as similarity measure						mean	variance
σ method	0.11	0.13	0.15	0.17	0.19	0.15	0.0001
MI	0.6923	1.2343	1.5187	1.2623	0.7232	1.0862	0.1317
MCC	1.4358	1.4492	1.5103	1.4438	1.4358	1.4550	0.0010

model using MCC as similarity measure						mean	variance
σ method	0.11	0.13	0.15	0.17	0.19	0.15	0.0001
MI	0.7928	0.8783	0.9892	0.9138	0.8283	0.8805	0.0058
MCC	0.9583	0.9738	0.9931	0.9727	0.9568	0.9709	0.0002

II. Kernel Method for Nonparametric Image Segmentation

Mumford-Shah (MS) model and CV model

- MS model (u is piecewise smooth):

$$\min_{C,u(x)} \int_{\Omega\backslash C} \alpha|\nabla u(x)|^2 dx + \int_{\Omega} (I(x) - u(x))^2 dx + \beta|C|.$$

- MS Cartoon / CV model (u is piecewise constant):

$$\min_{C,m_1,m_2} \sum_{i=1}^{2} \int_{\Omega_i} (I(x) - m_i)^2 dx + \beta \int_0^1 |C'(p)|dp.$$

Ω_1: inside C, Ω_2: outside C, $\Omega = \cup_{i=1}^{2}\Omega_i$.

Mumford and Shah ’89, L. Ambrosio et.al. ’97, Amadieu et. al. ’99, A. Tsai et.al.’01, T. Chan et. al. ’02 and A. Yezzi et.al.’02.

Parametric Active Contour Models for Segmentation

- Basic assumptions:
  (1). Image is composed by N disjoint regions

$$\Omega = \cup_i^N \Omega_i, \qquad \Gamma = \cup_i^N \partial\Omega_i.$$

  (2). Pixel intensity $\{I(x)\}$ for $x \in \Omega_i$ are i.i.d samples from a parametric pdf $p(I(x)|\theta_i)$.
- Joint pdf (Likelihood)

$$p(\{I(x), x \in \Omega\}|\{\theta_i\}) = \prod_i \prod_{x \in \Omega_i} p(I(x)|\theta_i).$$

- Maximum likelihood Estimation (MLE) & Maximum A Posteriori Estimation (MAP):

$$\min_{\{\theta_i\}} - \sum_i \int_{\Omega_i} \log p(I(x)|\theta_i) dx + \lambda|\Gamma|.$$

Image segmentation using Gaussian distribution model

- Gaussian distribution (e.g. region competition S.Zhu-A.Yuille 96 , geodesic active region N.Paragios-R.Deriche 02):

$$p(I(x)|\theta_i) = \frac{1}{(\sqrt{2\pi}\sigma_i)} e^{-\frac{|I(x)-\mu_i|^2}{2\sigma_i^2}}.$$

- Penalized MLE and MAP for segmentation:

$$\min_{C,\{\mu_i\},\{\sigma_i\}} \sum_i \int_{\Omega_i} \{\frac{1}{2\sigma_i^2}|I(x)-\mu_i|^2 + \ln \sigma_i\} dx + \beta|\Gamma|.$$

- Drawbacks: Prior knowledge on intensity distributions can be a significant restriction in real applications, especially, for images with high level of noise or complex multi-intensity distribution.

Non-Parametric Models for Segmentation

- Use of non-parametric density estimation is more accurate and flexible when the intensity distribution is unknown.
- Kernel method for pdf estimation: Let $\{x_1, x_2, ..., x_n\}$ be the independent samples drawn from $\widehat{p}(X)$.

$$\widehat{p}(X = x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i), \quad K_h(z) = \frac{1}{h^d}K(\frac{z}{h}).$$

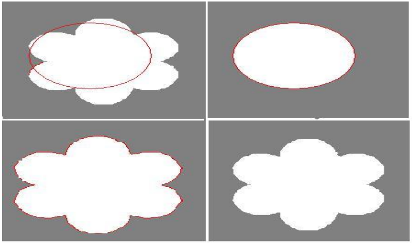
  K is a window function: smooth, positive, rapidly decreasing to zero outside the window, $\int_R K(z)dz = 1$. h is the band width.
- Using $\widehat{p}(I(x), \Omega_i) = \frac{1}{|\Omega_i|}\int_{\Omega_i} K_h(I(x) - I(y))dy$ for segmentation.
- Parzen Density Estimate: $K_h(z) =: G_\sigma(z) = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{|z|^2}{2\sigma^2}}$.

Nonparametric segmentation using MCC

- Label image L: Represent the position of a contour C in image I:

$$L(x) = \begin{cases} c_1 & \text{if } x \in \Omega_1; \\ c_2 & \text{if } x \in \Omega_2 = \Omega \setminus \Omega_1. \end{cases}$$

- For binary image I, $CC(I, L)$ is maximized if C gives a correct segmentation.
- For general image I, $MCC(I, L)$ (i.e. $CC(f(I), L)$) is maximized if C gives a correct segmentation.



MCC segmentation model:

- Problem: Find a smooth contour C separating Ω into two disjoint regions.
- Model: find C and f in RKHS, s.t. $CC(f(I), L)$ is maximized

$$\min_{C,f} E(C, f) = \min_{C,a_i} E(C, a_1, \cdots, a_n) = \oint_C \mathrm{d}s + \frac{\lambda}{2}(1 - CC(f(I), L))^2.$$

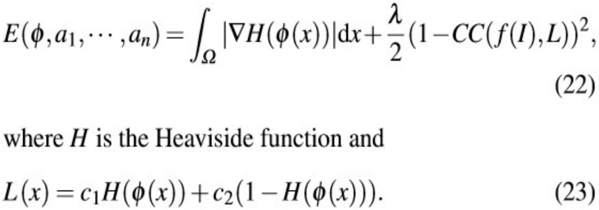
where

$$f(I(x)) = \sum_i^n a_i K_h\left(I(x), \xi_i\right),$$

$$CC(f(I), L) = \frac{Cov(f(I), L)}{\sqrt{Var(f(I))}\sqrt{VarL}},$$

$$Cov(f(I), L) = E[(f(I) - E(f(I)))(L - E(L)].$$

MCC segmentation model - level set formulation & Soft model



$$E(\phi, a_1, \cdots, a_n) = \int_\Omega |\nabla H(\phi(x))| \mathrm{d}x + \frac{\lambda}{2}(1 - CC(f(I), L))^2, \tag{22}$$

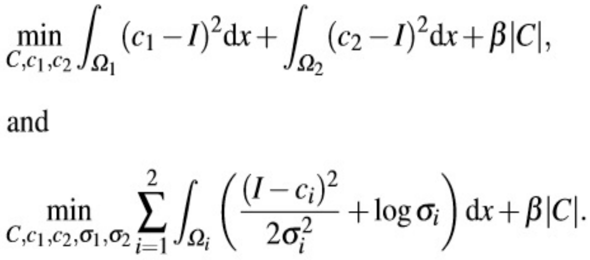
where H is the Heaviside function and

$$L(x) = c_1 H(\phi(x)) + c_2(1 - H(\phi(x))). \tag{23}$$

Soft MCC model: Replace $H(\phi(x))$ by $u(x)$, $0 \leq u(x) \leq 1$.

Compared models - CV and parametric models

- $\Omega = \Omega_1 \cup \Omega_2, \;\; \Omega_1 \cap \Omega_2 = \emptyset.$
- CV model: $I(x) = C_i$ for $x \in \Omega_i$.
- parametric model: $I(x) \sim G(C_i, \sigma_i)$ for $x \in \Omega_i$.

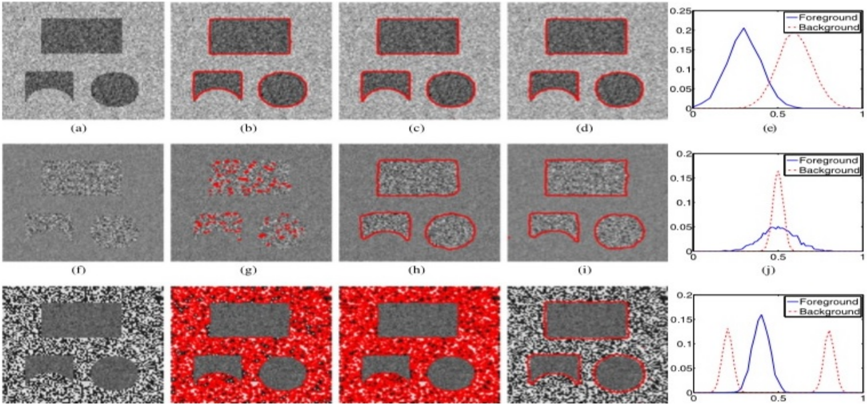


$$\min_{C,c_1,c_2} \int_{\Omega_1} (c_1 - I)^2 \mathrm{d}x + \int_{\Omega_2} (c_2 - I)^2 \mathrm{d}x + \beta |C|,$$

and

$$\min_{C,c_1,c_2,\sigma_1,\sigma_2} \sum_{i=1}^{2} \int_{\Omega_i} \left(\frac{(I - c_i)^2}{2\sigma_i^2} + \log \sigma_i \right) \mathrm{d}x + \beta |C|.$$

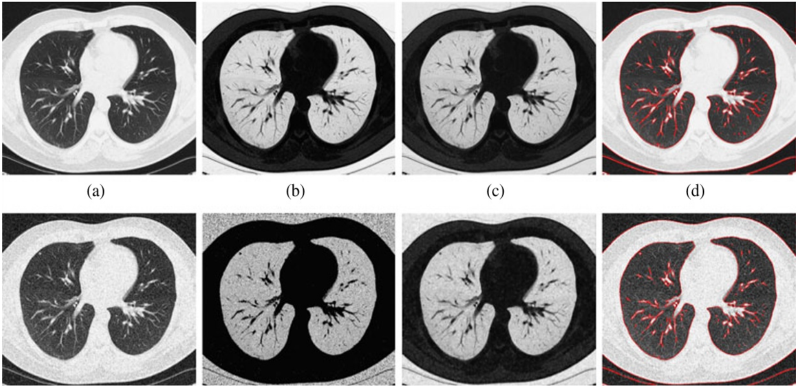
Comparisons - CV, Gaussian and MCC models


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

(f) (g) (h) (i) (j)

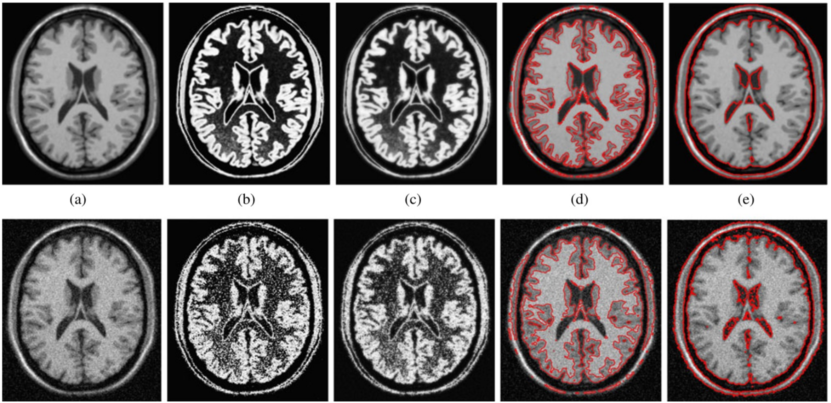
Columns L to R: I, $CV(C_1, C_2)$, $G(C_i, \sigma_i)$, $MCC(I, L)$, distribution of I.

MCC model for lung image segmentation



Columns L to R: I, $f(I)$, L, $u = 0.5$ superimposed on I.
Row top to bottom: results for clear image to noisy image.

MCC / Gaussian models for brain image segmentation



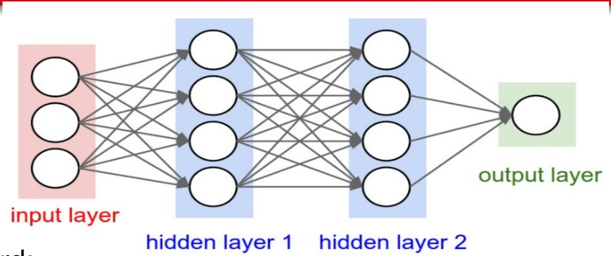
Columns L to R: I, $f(I)$, L, seg from $MCC(I, L)$, seg from $G(C_i, \sigma_i)$.
Rows top to bottom: results for clear image to noisy image.

Conclusion on kernel method for image analysis (part I, II)

By using kernel method we gain the ability to implement nonlinear models in the input space by linear modeling in the RKHS.

III. Kernel Method in Deep Learning

Artificial neuron network (ANN), Multi-layer Perceptron (MLP)



- Feedforward:

$$h^1 = f(W^0 x + b^0), \quad \dots \quad h^{m+1} = f(W^m h^m + b^m)$$

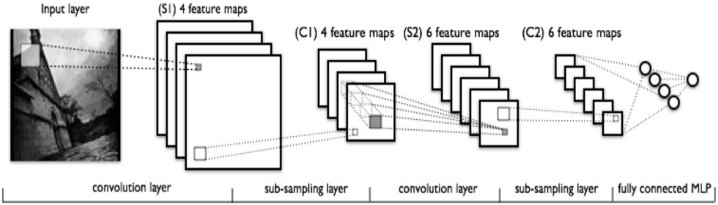
$$f(z) : \tanh z = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \; sigmoid \; z = \frac{1}{1 + exp\{-z\}}, \; softmax = max\{0, z\}$$

- backpropagation:

$$\min_{W,b} L(g(x, W, b), y).$$

$$g(x, W, b) = f(W^m(f(W^{m-1} \dots f(W^0 x + b^0) \dots)) + b^m)$$

Convolutional neuron network (CNN) (Ruslan Salakhutdinov)

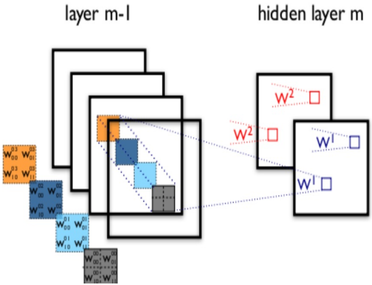


$$h^1_{i,j} = f((W^0 * x)_{i,j} + b^0), \quad \dots \quad h^{m+1}_{i,j} = f((W^m * x)_{i,j} + b^m)$$

.

$$g(x, W, b) = f(W^m(f(W^{m-1} \dots f(W^0 * x + b^0) \dots)) + b^m).$$

Convolutional neuron network (CNN) / Conv. operator



- Filter/kernel W^k is a 3d tensor, $W^k := W_{i,j}^{k,l}$.
- k: the k-th feature map at layer m, l: the l-th feature map at layer $(m-1)$, (i,j): pixel coordinates.
- In the graph, hidden layers $m-1$ and m contain 4 and 2 feature maps, resp, i.e. $k = 1, 2$ and $l = 1, 2, 3, 4$.

Integration of deep neural networks and kernel methods

- Deep neural networks (e.g. MLP, CNN)
  - Scalable
  - Nonlinear function f at each layer is simple, but multi-layers are required to approximate given data.
  - Output $g(x, W)$ is non-convex in W . Non-convex optimization, local minima for backpropagation..
- Kernel methods
  - Universal approximation property (strictly PD kernel): For any continuous mapping $g(x)$,

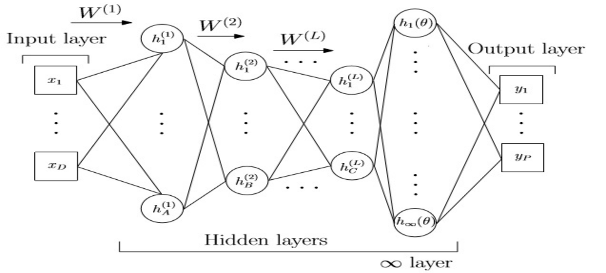
$$g(x) = g(x, \alpha) = \lim_{m} \sum_{i=1}^{m} K_\theta(x_i, x)\alpha_i, \quad in \ \ L^p.$$

  - The approximation is always centered on the available data, which simplifies representation and training procedures.
  - $g(x, \alpha)$ is linear in α. $\min_\alpha L(g(x, \alpha), y)$ can be convex optimization.
  - Computation cost increases as the data increases.

Integration of deep neural networks and kernel methods

- Common feature of deep neural networks and kernel methods: Both can learn universal nonlinear mappings directly from data for regression, classification, prediction.
- Can we integrate the convexity of kernel methods with the scalability of deep neural networks for better learning?

Deep kernel learning (A.G.Wilson et al. Deep kernel learning 2015)



- Hidden layers 1-L: (e.g. CNN), parameters $(W^1, \dots, W^L)$;
  The last hidden layer: spectral mixture kernels for a Gaussian process:

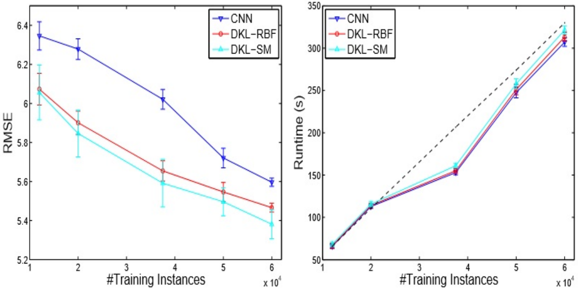
$$K_{sm}(x, x'|\theta) = \sum_q a_q G(x - x', \Sigma_q) \cos < x - x', 2\pi\mu_q > .$$

- kernel design for a Gaussian process in deep architectures:

$$K_{sm}(x_i, x_j|\theta) = K(g(x_i, W), g(x_j, W)|(\theta, W)).$$

Deep kernel learning (DKL) experiment

- DKL: 2 layers CNN + 4 full connected layers + kernel method
- comparisons of CNN vs. DKL



More thoughts? Feature work

- Design kernels in deep RKHS architectures (more powerful than multi-scale kernels)?
- Modification of hierarchical systems that allow bidirectional information transfer and/or activations in hidden layers?
- Better training, scalable and ease of implementation learning algorithms for practical use?

Thanks