Topics:

1. Foundations of Neuroanatomy
   - Brain Axes
   - Lobes of Cerebrum
   - Homunculus
   - Structure of Neurons

2.1 Fourier Analysis
   - Fourier theorem for periodic functions
   - Scalar Product for 2π-periodic functions
   - Computing coefficients for sinusoidal decomposition
   - Gibbs Phenomenon
   - Euler's formula
   - Fourier transformation
   - Specific Fourier transforms (box $\leftrightarrow$ sinc, dirac $\leftrightarrow$ constant, comb, gaussian)
   - Convolution
   - Convolution theorem
   - Intuition of fourier transform of images

2.2 Signal Processing
   - Mathematical Model of Sampling
   - Aliasing
   - Nyquist theorem

2.3 Image Resampling
   - Sampling tasks
   - Nearest-neighbor
   - linear interpolation
   - Cubic Spline interpolation
   - Cubic B-spline interpolation
   - image pyramids

3. Magnetic Resonance Imaging
   - Safety issues
   - Structure of MRI scanner (main coil, gradient coil, transceiver)
   - Quantum mechanical spin (magnetic moment)
   - Nuclear Zeeman effect
   - Equilibrium magnetization
   - Larmor precision
   - Magnetic resonance
   - Relaxation (longitudinal / transverse)
   - T_1 decay
   - T_2 decay
   - T_2^* decay
   - time of echo
   - time of repetition
   - Spin echo
   - Gradient recalled echo
   - Slice selection
   - Slice refocusing
   - frequency encoding
   - phase encoding
   - k-space
   - Echo-Planar Imaging
   - RF noise
   - Gibbs ringing
   - EPI artifacts "N/2 ghosts"
4. Registration and Normalization
   - Registration definition
   - Registration types
   - Registration evaluation
   - Types of affine transformations
   - Coordinate types
   - Cost functions for registration
   - Entropy / information
   - Optimization of transforms
   - Optimization types
   - Downhill Simplex method
   - Powell's method
   - Warp fields
   - Cardinal B-spline
   - Gradient descent
   - Newton's method
   - Gauss-Newton method
   - Levenberg's method
   - Levenberg-Marquardt's method
   - Regularizing non-linear registration
   - Pasha algorithm

- Optical flow
- Demons algorithm
- Diffeomorphism
- Velocity field representation
- Talairach and MNI templates
- Iterative Template Refinement

5. Segmentation

- Segmentation task
- Segmentation evaluation
- Deformable contours
- Optimizing a contour
- k-means for segmentation
- Gaussian mixture models
- Expectation Maximization for GMM
- Markov Random Fields
- Hammersley-Clifford theorem
- HMRF-EM algorithm
- ICM method
- Min-cut / Max-flow for MRF
- Alpha expansion
- Bias fields

1. Fourier Analysis and Signal Processing

Definition: Fourier Series

$$f(x) = \sum_{n=0}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$
$$= \sum_{n=0}^{\infty} \left(a_n \frac{e^{inx} + e^{-inx}}{2} - i b_n \frac{e^{inx} - e^{-inx}}{2} \right)$$
$$= \sum_{n=0}^{\infty} \left(\frac{a_n - i b_n}{2} e^{inx} + \frac{a_n + i b_n}{2} e^{-inx} \right) = \sum_{n=-\infty}^{\infty} c_n e^{inx}$$

Theorem: Fourier Theorem for Periodic Functions

For a 2π periodic, piecewise monotone, bounded real function $f(x)$, values at points of continuity x equal those of its fourier series:
→ indication how to find coefficients

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

Computing Coefficients

a_0: $a_0 = \frac{1}{2\pi}\langle f, u_0 \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \cos(0x)\, dx = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx$ from:

$$\langle f, u_0 \rangle = \left\langle \sum_{n=0}^{\infty} a_n u_n + b_n v_n \,,\, u_0 \right\rangle$$
$$= \langle a_0 u_0, u_0 \rangle$$
$$= a_0 2\pi$$

a_m: $a_m = \frac{1}{\pi}\langle f, u_m \rangle = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(mx)\, dx$ from:

$$\langle f, u_m \rangle = \left\langle \sum_{n=0}^{\infty} a_n u_n + b_n v_n \,,\, u_m \right\rangle$$
$$= \langle a_m u_m, u_m \rangle$$
$$= a_m \pi$$

Definition: Scalar Product on 2π-periodic Functions

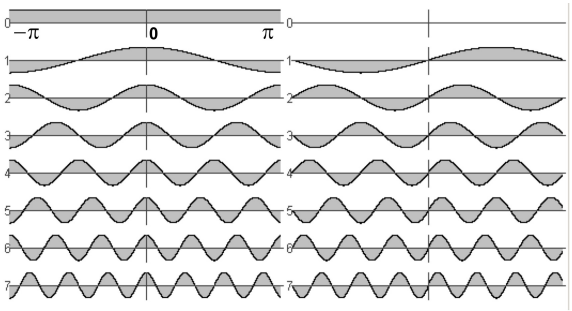
$$\langle f, g \rangle := \int_{-\pi}^{\pi} f(x) g(x)\, dx$$

Orthogonality of Sines and Cosines

$$\langle u_n, u_m \rangle = \int_{-\pi}^{\pi} \cos(nt) \cos(mt)\, dt = \begin{cases} 0 & \text{if } m \neq n \\ \pi & \text{if } m = n = 1,2,\dots \\ 2\pi & \text{if } m = n = 0 \end{cases}$$

$$\langle v_n, v_m \rangle = \int_{-\pi}^{\pi} \sin(nt) \sin(mt)\, dt = \begin{cases} 0 & \text{if } m \neq n \\ \pi & \text{if } m = n = 1,2,\dots \\ 0 & \text{if } m = n = 0 \end{cases}$$

$$\langle u_n, v_m \rangle = \int_{-\pi}^{\pi} \cos(nt) \sin(mt)\, dt = 0 \quad \forall n, m = 0,1,2,\dots$$

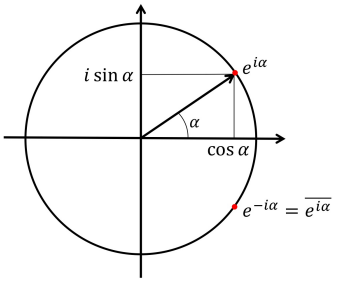


$u_n(t) := \cos(nt), n = 0,1,2,\dots$ $v_n(t) := \sin(nt), n = 0,1,2,\dots$

Euler's formula: $e^{i\alpha} = \cos\alpha + i \sin\alpha$

Cosine: $\cos\alpha = \frac{e^{i\alpha} + e^{-i\alpha}}{2}$

Sine: $\sin\alpha = -i \frac{e^{i\alpha} - e^{-i\alpha}}{2}$



Theorem:

A fourier series is only real, if: $c_n = \overline{c_{-n}}$

Definition: Fourier Transform

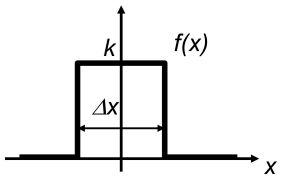
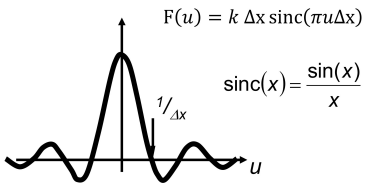
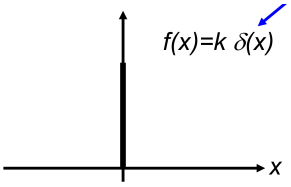
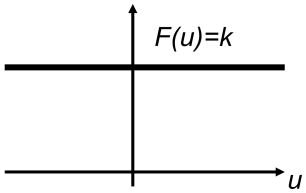
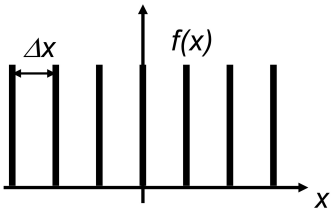
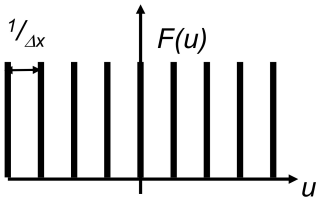
Inverse Transform: $f(x) = \int_{-\infty}^{\infty} F(u)\, e^{2\pi i u x}\, du$

Forward Transform: $F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i u x}\, dx$

Theorem: Fourier Duality

If F(u) is the Fourier transform of f(x), f(-x) is the Fourier transform of F(u)

Fourier Pairs

Spatial Domain	Frequency Domain
Box: 	Sinc: 
Dirac delta: 	Constant: 
Comb: 	Comb: 

Fourier Properties:

Property	Time / Space	Frequency Domain
Linearity	$c_1 \cdot f(x) + c_2 \cdot g(x)$	$c_1 \cdot F(u) + c_2 \cdot G(u)$
Time scaling	$f(c \cdot x)$	$\frac{1}{\lvert c \rvert} F\left(\frac{u}{c}\right)$
Space/Time shift	$f(x-x_0)$	$F(u)e^{-i2\pi u x_0}$
Time derivative	$df(x)/dx$	$i2\pi u \cdot F(u)$
Complex conjugation	$\overline{f(x)}$	$\overline{F(-u)}$
Even signals	$f(-x)=f(x)$	$F(-u)=F(u)$
Odd signals	$f(-x)=-f(x)$	$F(-u)=-F(u)$
Convolution	$f(x)*g(x)$	$F(u) \cdot G(u)$
Modulation	$f(x) \cdot g(x)$	$F(u)*G(u)$

Definition: Convolution

$$f(x) = (h * g)(x) = \int_{-\infty}^{\infty} h(\xi) \cdot g(x - \xi) d\xi$$

Theorem: Convolution Theorem

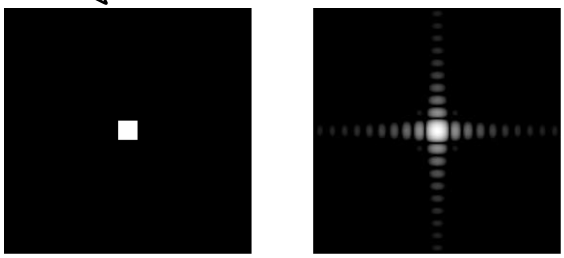
$f(x)=g*h$ corresponds to $F(u)=G \cdot H$ and $f(x) = h(x) \cdot g(x)$ corresponds to $F(u) = H(u) * G(u)$

Fourier Image Pairs

Circle:

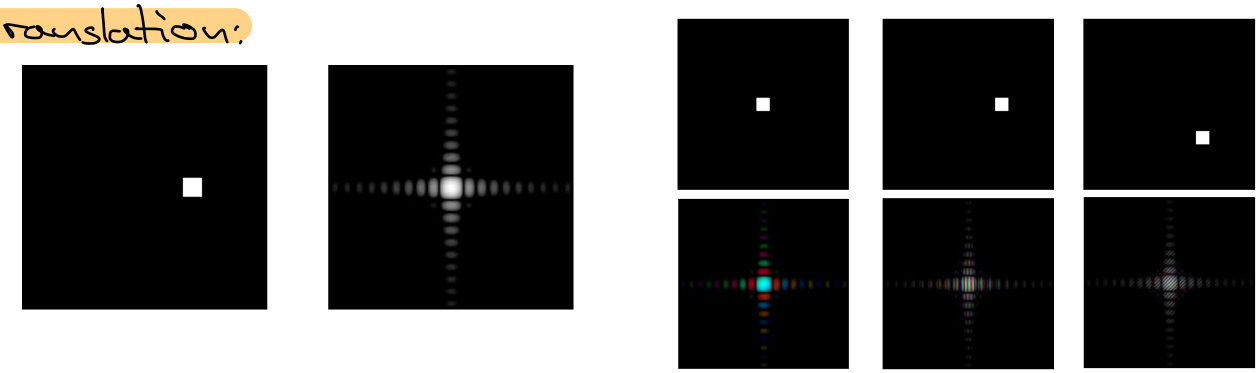
Square:

Scaling:

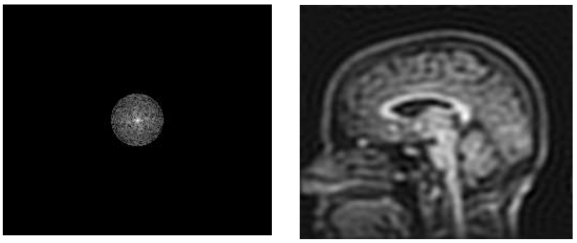


Rotation:

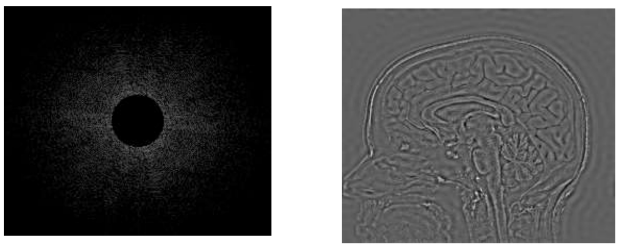
Translation:



Low-pass:



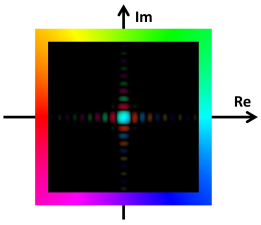
High-pass:



Fourier Color Coding

Brightness: magnitude

Color: phase



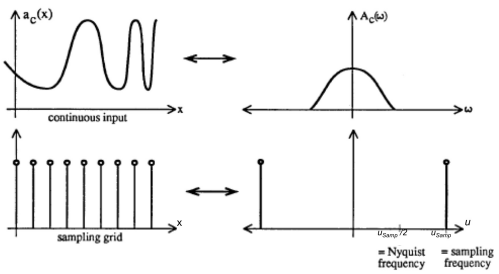
Mathematical Model of Sampling

Sample the continuous signal in discretized steps

Sampling function: comb

Sampling: $f_s(x) = f(x) \amalg_{\Delta x} = F(u) * \amalg_{1/\Delta x}$

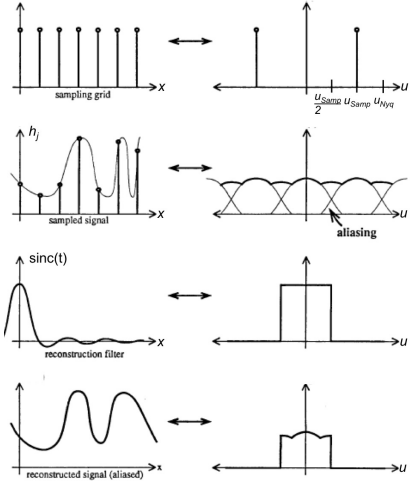
→ copies frequency spectrum along frequency domain with distance $1/\Delta x$



Reconstruction: Cut out frequencies with box function which equals to the convolution with the sinc function in the spatial domain

Definition: Aliasing

Undersampling leads to aliasing which induces new frequencies due to overlapping frequencies when sampling



Theorem: Nyquist Theorem

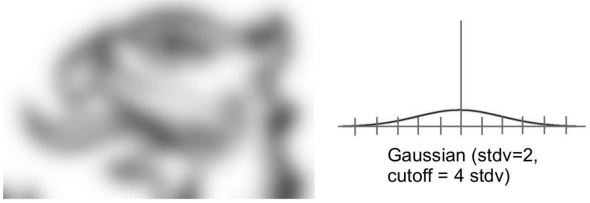
If our original signal is band-width limited we can prevent aliasing, by sampling at double the frequency than the maximal observed frequency u_{max}: $\omega \geq 2u_{max}$
→ Nyquist frequency

Convolution-based Reconstruction

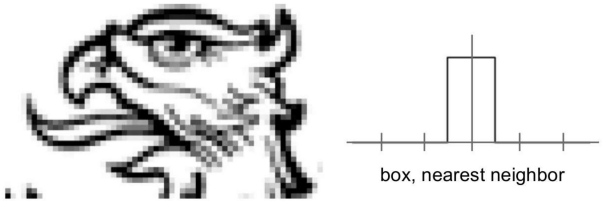
A continuous signal can be interpolated by convolving with some kernel
→ kernel has to sum to 1
→ keep support minimal for efficiency

Hann-windowed Sinc:

Gaussian:

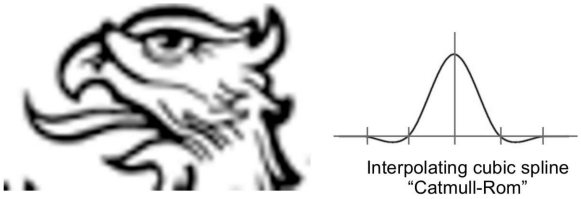


Box:



Linear:

Interp. Cubic Spline:



Non-interp. Cubic B-spline:

Boundary Padding:

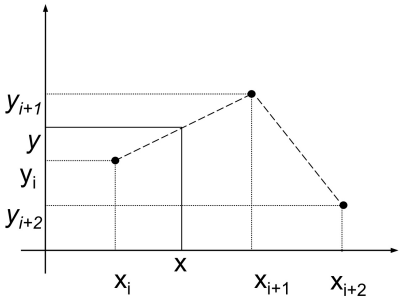
1. Cut-off / ignore
2. Dirichlet: Fixed value padding
3. Neumann: Mirror padding
4. Periodic: Copy from other side

Piecewise Linear Interpolation

$$f(x) = (1-\alpha)y_i + \alpha y_{i+1}$$

where

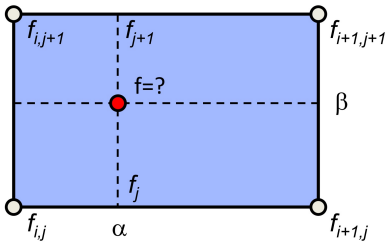
$$\alpha = \frac{x-x_i}{x_{i+1}-x_i} \in [0,1]$$



Bilinear Interpolation

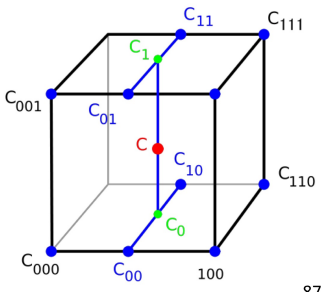
$$f_j = (1-\alpha)f_{i,j} + \alpha f_{i+1,j} \qquad f(x,y) = (1-\beta)f_j + \beta f_{j+1}$$

$$f_{j+1} = (1-\alpha)f_{i,j+1} + \alpha f_{i+1,j+1} \qquad \alpha = \frac{x-x_i}{x_{i+1}-x_i}, \qquad \beta = \frac{y-y_i}{y_{i+1}-y_i},$$



Trilinear Interpolation

Generalization of bilinear interpolation to 3D



3. Magnetic Resonance Imaging

Basic Principles

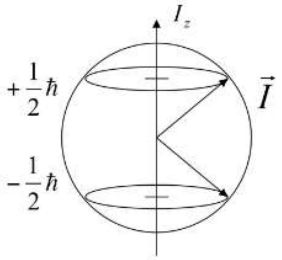
Quantum Mechanicals Spin

Subatomic particles have an elementary property spin which is similar to angular momentum. These are fixed quantities and typically cancel out for atoms. Since the hydrogen atom consist of a single proton it has a net magnetization.

Proton spin: $I_Z = \pm\frac{1}{2}\hbar$

h = Planck constant ≈ $6.626068 \cdot 10^{-34}$ Js

$\hbar = \frac{h}{2\pi}$ "reduced Planck constant"



→ induces net magnetization
→ proton can be in lower and higher energy state

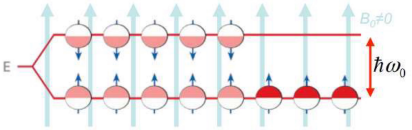
Nuclear Magnetic Moment: $\boldsymbol{\mu = \gamma I}$

→ spin induces magnetization

I: spin
γ: nucleus-specific gyromagnetic ratio

Nuclear Zeeman Effect

When exposed to an external magnetic field B_0, the z-components of the nuclear magnetic moments align with B_0



Moments either align in parallel or anti-parallel state:

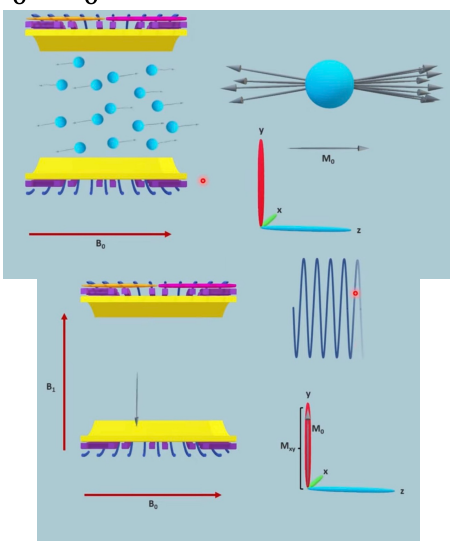
$$E_{\uparrow,\downarrow} = \mu_{\uparrow,\downarrow} B_0 = \mp\frac{1}{2}\hbar\gamma B_0$$

Zeeman splitting: $\Delta E = \hbar\gamma B_0$

Equilibrium magnetization: tiny excess of spins in lower energy state
→ induces net magnetization in tissue $M_0 \sim B_0$

Boltzmann statistics: $\frac{n_\downarrow}{n_\uparrow} = e^{-\frac{\Delta E}{k_B T}}$

→ excess of spins in lower energy state



Larmor Precession

Precession frequency of the nuclei spin around the direction B_0.

$$\boxed{\omega_0 = \gamma B_0}$$

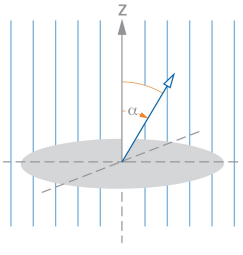
→ out-of-phas spins eliminate transversal component

Nuclear Magnetic Resonance

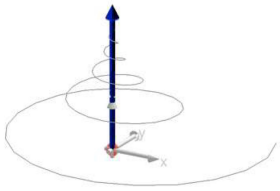
An electromagnetic RF pulse perpendicular to B_0 at the Larmor frequency of given tissue lets the spins spin in frequency an the magnetization vectors align with the pulse moving into the transversal plane

Flip angle: $\alpha = \gamma \int_{-\tau/2}^{\tau/2} B_1(t)dt$

→ angle of flipped magnetization vector

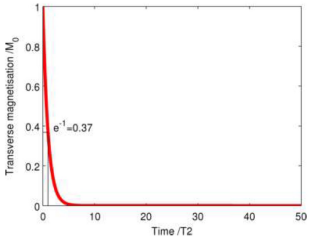


Relaxation



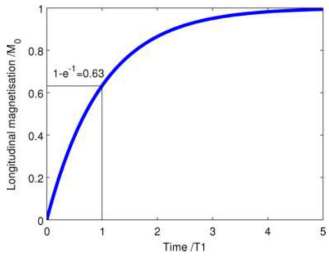
After the pulse the magnetization returns to the equilibrium state

Transversal: Spin-spin interactions cause the spins to go out of phase until they cancel out the transversal magnetization



Evolution: $M_{xy} = M_0 e^{-\frac{t}{T_2}}$

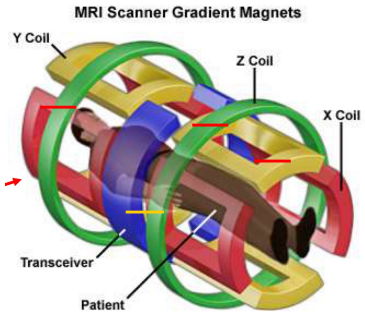
Longitudinal: Spin-lattice interaction cause the nucleis to lose energy and align back with B_0 resulting in a gain of longitudinal magnetization



Evolution: $M_z(t) = M_0 + (M_z(0) - M_0)e^{-\frac{t}{T_1}}$

Assuming 90° at t=0: $M_z = M_0(1 - e^{-\frac{t}{T_1}})$

MR Scanner Components



Main Coil: Induces the main magnetic field

Gradient Coils: Induce gradients in the magnetic field to select slices and encode x,y coordinates

RF Transmitter: Emit RF pulse resonanced within a slice of the patient

Transceiver: Sample the emitted magnetic resonance at discrete steps

MR Safety issues

Main field:

- permanently cooled (<4K) superconductors
- huge force on ferromagnetic materials
- electronic devices may be destroyed
- induced magnetic flux may cause dizziness

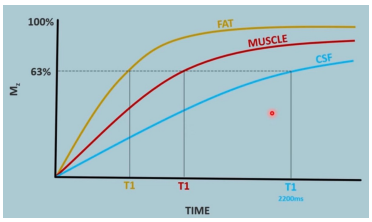
Field Gradients:

- patients should not cross hands or feet to not create a conducting loop
- rate of gradient is limited to not overstimulate patient's peripheral nerves

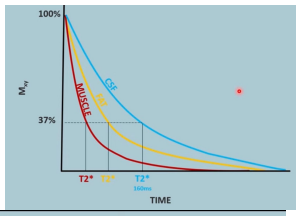
RF Fields:

- hot spots on metallic materials
- high frequencies required for strong B_0

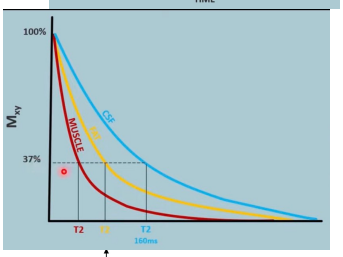
MR Imaging Basics



T_1 time: Spin-lattice interaction
⇒ gain of longitudinal magnetization
→ measured at 63% gain of magnetization

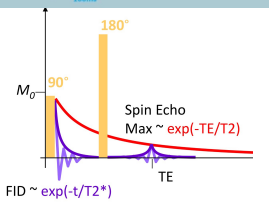


T_2^* time: Spin-spin interaction & field inhomogeneities
⇒ loss of transversal magnetization
→ typically measured and has to be accounted for
→ field inhomogeneities induce faster dephasing



T_2 time: Spin-spin interaction
⇒ loss of transversal magnetization
→ measured at 63% loss of magnetization

T_2^* compensation: Apply 180° pulse to invert spins which induces a rephasing of the spins
→ more dephased atoms contribute more to magnetization which cancels out unwanted effects and aligns with T_2 time



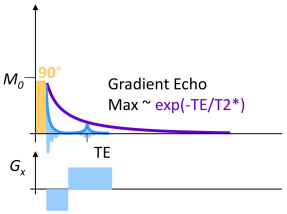
Time of Echo (TE): time of/around measurement

Time of Repetition (TR): duration between two RF pulses
→ determines how far the longitudinal component can recover

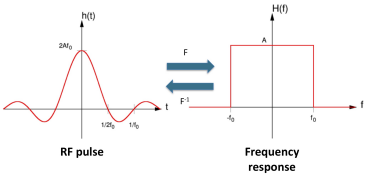
Spin Echo: Time of echo after rephasing where spin becomes in phase and equals T_2 time

Gradient Recalled Echo

Echo induced by dephasing and subsequent rephasing.
→ faster and less RF than spin echo
→ does not compensate dispersions induced by B_0



RF Pulse: If we want to excite a rectangular slice profile in frequency domain, we have to convolve with Sinc in the time domain



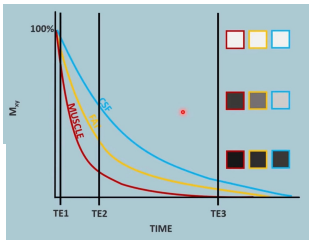
Characteristic times

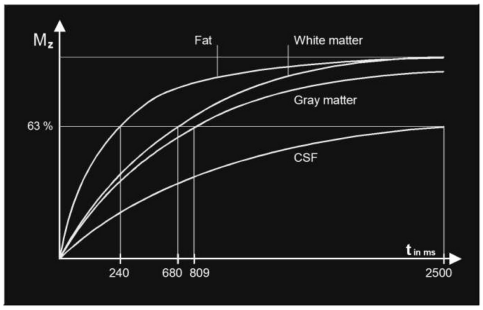
T_1:

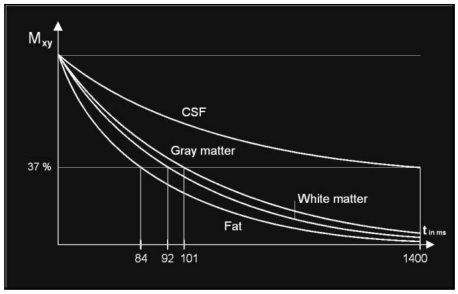
	0.2 T	1.0 T	3T
Fat		240	≈400
Muscle	370	730	≈1200
White Matter	388	680	≈900
Gray Matter	492	809	≈1500
CSF	1400	2500	≈4000

T_2:

Fat	84
Muscle	47
White Matter	92
Gray Matter	101
CSF	1400



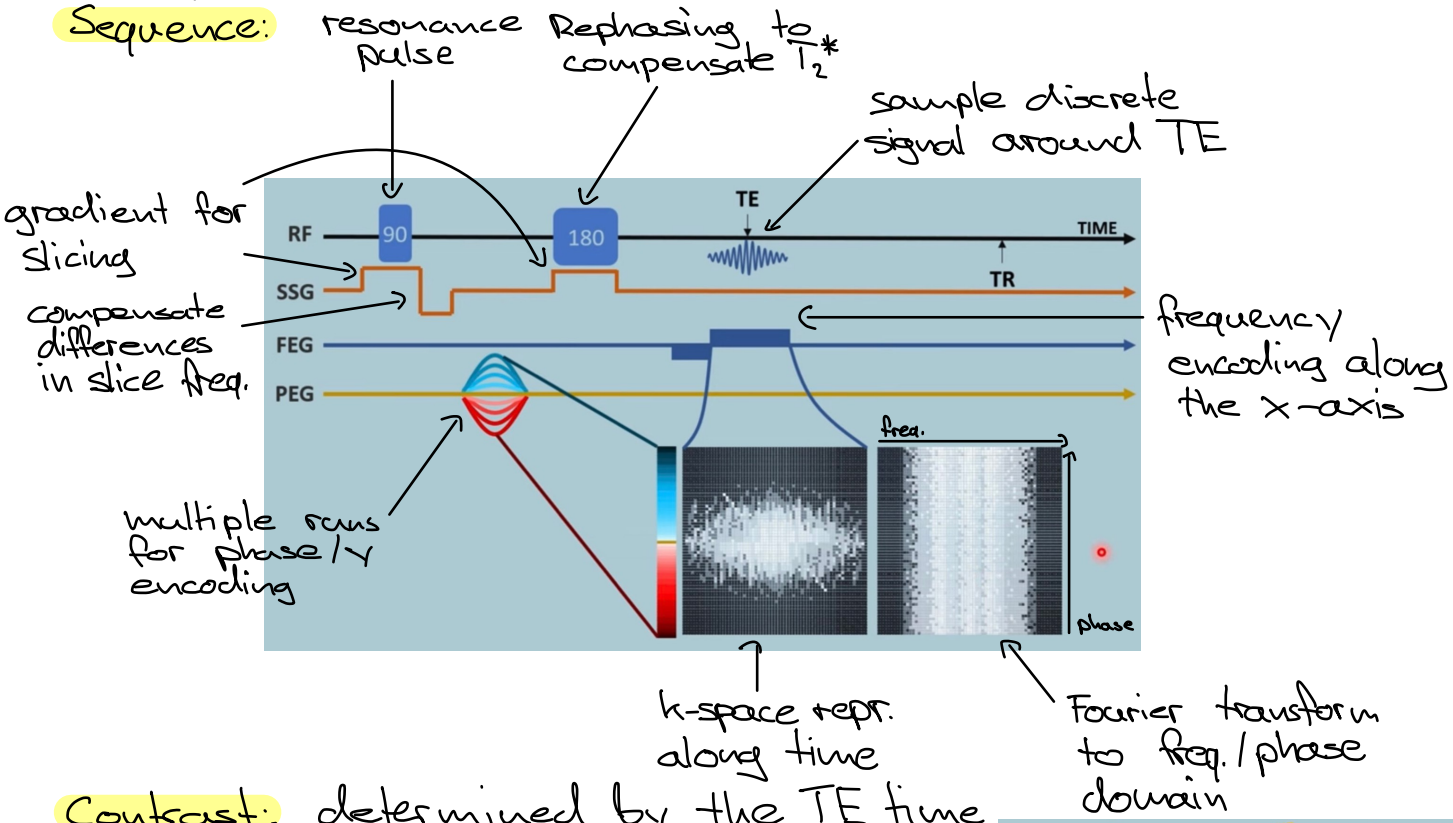




MR Imaging

T_2 Imaging: Measure contrast in T_2 times

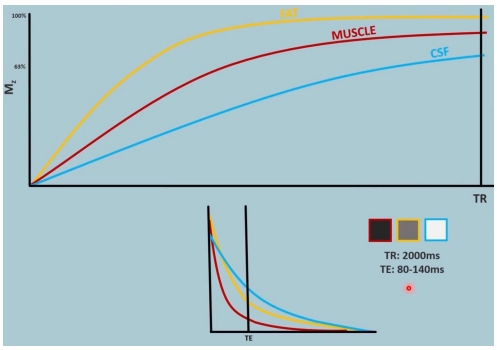
Sequence:



Contrast: determined by the TE time

Characteristics:

- **Long TE:** allow difference in decay
- **Long TR:** allow full recovery in longitude for full transversal magnitization



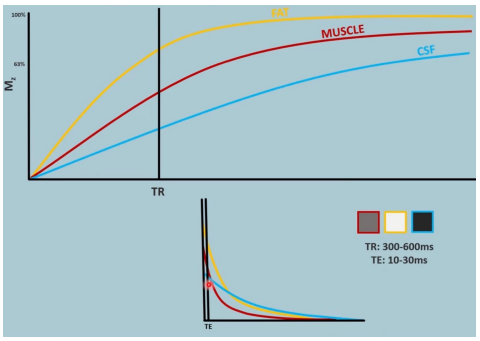
T_1 Imaging:

Sequence:

Contrast: determined by TR time

Characteristics:

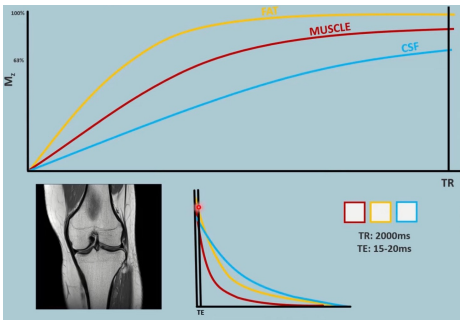
- **Short TE:** no influence of T^2 decay
- **Short TR:** Do not let longitudinal comp. recover fully which leads to diff. transversal magnetization linked to the speed of recovery of each tissue material



Proton Density: Measure proton density

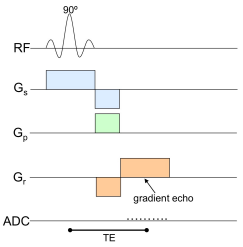
Characteristics:

- **Long TR:** full recovery of longitude
- **Short TE:** signal differences in magnitude can only come from diff. densities of protons

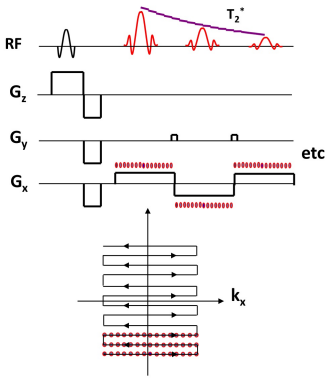


Gradient Echo:

Measure T_2^* time through echo coming from dephasing and phasing the spins through the gradient coils



Echo-Planar Imaging



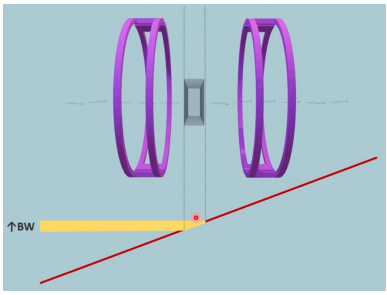
Continuous readout through a number of rephasing gradients instead of 180° pulse while changing the frequency encoding gradient to receive GRE.

→ allows to read out range of phases in a single TR

→ One-shot MR scan limits artifacts through patient's motion

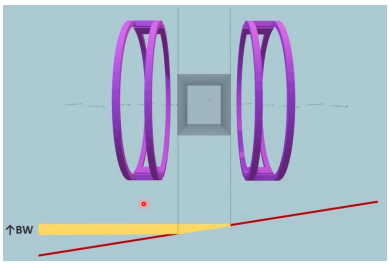
Slice Selection:

Induce gradient and select slice by using a RF pulse of a frequency corresponding to the Larmor frequency at the intended position



Position: changed either through field strength or pulse frequency

Thickness: $\Delta z = \frac{2\pi\Delta f}{\gamma G_z}$



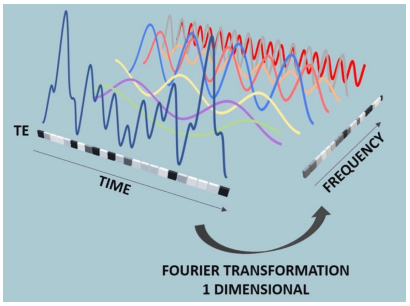
→ changed bandwith of the RF pulse or strength of gradient

Slice Refocusing:

Compensate for different Larmor frequencies within the slice by applying an inverse gradient

Phase dispersion: $\exp\left(\frac{i\gamma G_z \tau}{2}\right)$

τ: duration of pulse



Frequency Encoding:

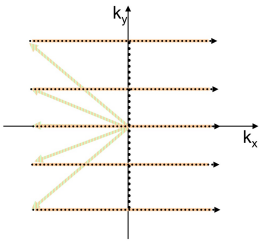
Encode the x-dimension through different frequencies which can be recovered through Fourier analysis

Sampling resolution: $\Delta x = FOV / N$ with: $f_{max} = \frac{\gamma G_x}{2\pi}\frac{FOV}{2} < \frac{1}{2\Delta t} \Rightarrow FOV = \frac{2\pi}{\gamma G_x \Delta t}$

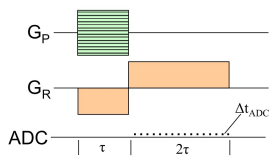
Phase Encoding:

Encode y-dimension through the phase by applying a gradient bringing the spin out of phase along the y-axis

→ one scan per phase encoding



k-Space: Space of stacked measurements per phase-shift over time



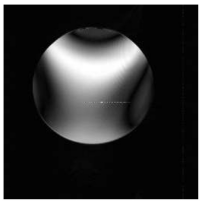
→ Size determined by discretization of ADC

→ technically suffices to sample half of k-space along y-direction but complete signal improves SNR

$$k_x = \frac{\gamma}{2\pi}\left(-G_x\tau + G_x\{0, \Delta t_{ADC}, 2\Delta t_{ADC}, \ldots, 2\tau\}\right)$$

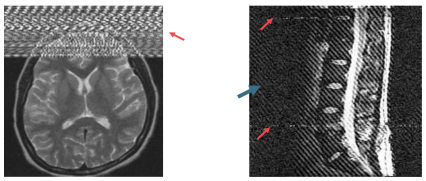
$$k_y = \frac{\gamma}{2\pi}\{G_{y,\max}, \ldots, G_{y,\min}\}\tau$$

MR Artefacts

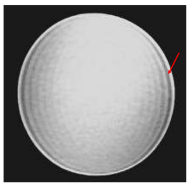


Inhomogeneous Field: Different field strengths

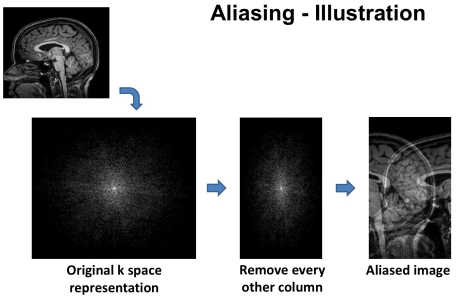
RF Noise: Imperfect pulse resulting in inaccurat excitation



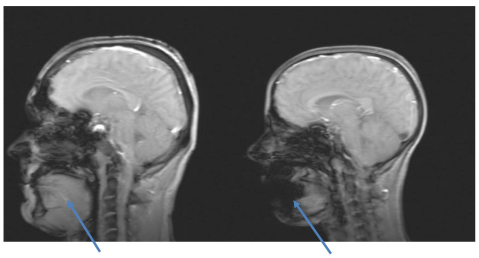
Gibbs Ringing: Truncation artefact due to finite sampling
→ prevented through denser sampling



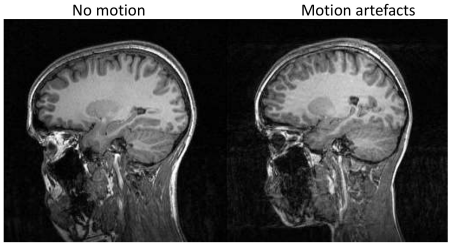
Aliasing: Fold-over from image due to too sparse sampling



Metal Artefacts: strong distortion of B_0
→ T_2^* strongly shortened
→ strong local image distortions

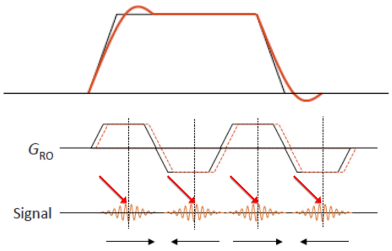


Motion Artefacts: Measurement of diff. k-spaces due to motion
→ blurring or repetion of structures



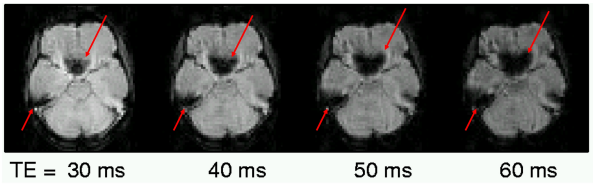
EPI Artefacts

N/2 Ghost: gradient shape is modified due to currents which shifts signal echo away from k-space centre
→ is redundant in lines of inverse phase
→ artefact is N/2 periodic in image

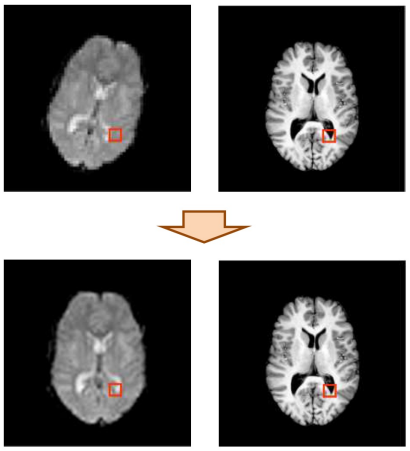


Susceptibility Artefacts

local changes in the magnetic field due to diff. magnetic properties of tissue
→ results in signal loss or distortions
→ especially visible in T_2^*-weighted seq.
→ reduce by combining information from both phase polarities



4. Registration and Normalization



Definition: Registration
Bring different images into spatial alignment either along time, between different modalities or between different subjects

Reference image: remains unchanged and is optimized to

Floating image: image being transformed

Registration Types

Affine Registration: only permits selected affine transformations

Nonlinear Registration: allow local deformations
→ often required after linear registration

Typical Assumptions:
- affine transformations have been accounted for
- commonly performed intra-modally

Intra-modality: Compare images coming from same modality.
→ intensities can be compared directly

Inter-modality: Compare images between modalities

Optimization-based: Iteratively optimize over space of transformations using a cost function

Derivative-free: Function $f(x)$ is only evaluated at points x

Derivative-based: Assumes $f(x)$ is smooth and computes/approximates a derivative

Local: Given an intial estimitate find a local minimum

Global: Find global minimum → in general not possible

Regression-based: Directly map two images to transformation parameters
→ typically learned through NN using opt.

Evaluation Techniques

Landmark-based

Compute the error between manually or anatomical designed landmarks.

Target Registration Error:

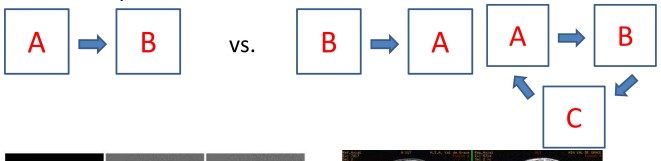
$$\mathrm{TRE}_{\mathrm{mean}} = \frac{1}{m}\sum_{i=1}^{m} \|\mathbf{r}_i - T(\mathbf{p}_i)\|$$

$$\mathrm{TRE}_{\mathrm{max}} = \max\{\|\mathbf{r}_i - T(\mathbf{p}_i)\| \mid i = 1, \ldots, m\}$$

Consistency-based:

Assess consistency of the compute transform

- Check inverse transform
- Check cyclic transformation
- Check transform with noise
- Check transform for re-scan



Visual Assessment

- Difference / average image
- Feature alignment
- Checkerboard patterns
- Color coding

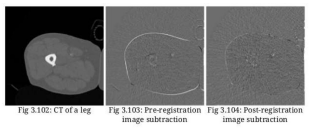
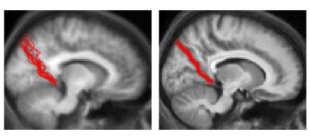
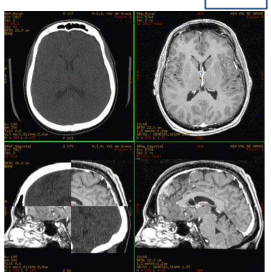
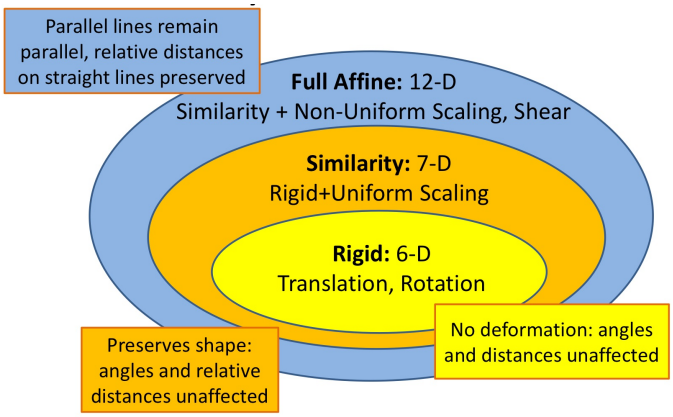


Fig 3.102: CT of a leg | Fig 3.103: Pre-registration image subtraction | Fig 3.104: Post-registration image subtraction





Affine Transformations



Homogenous Transformation:

Scaling + Rotation + Shear

$$\begin{pmatrix} a & b & c & tx \\ d & e & f & ty \\ g & h & i & tz \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Translation

Index / World Coordinates

Index coordinates indexing the memory are defined as:
origin: **p** basis: **(e,f,g)**

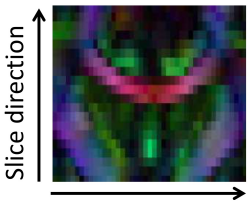
Transformation:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{p} + i\mathbf{e} + j\mathbf{f} + k\mathbf{g} \qquad \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} | & | & | & | \\ \mathbf{e} & \mathbf{f} & \mathbf{g} & \mathbf{p} \\ | & | & | & | \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \\ 1 \end{pmatrix}$$

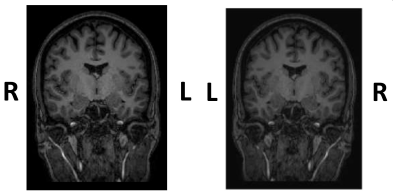
Isotropic: voxels ((e,f,g)) have same length

Anistropic: distortion of image


In-plane slice



Handedness:

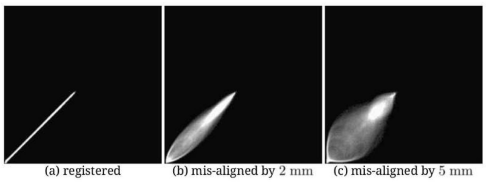


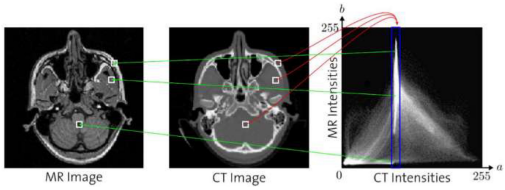
Radiological Neurological

Orientation: LAS = left-anterior-superior → left-handed
RAS = right-anterior-superior → right-handed

Histograms for Registration

Represent joint distribution of two images through the correlation of the pixel by displaying each along a single axis
- → well-aligned images have a peak along the diagonal
- → mapping might not be one-to-one between modalities

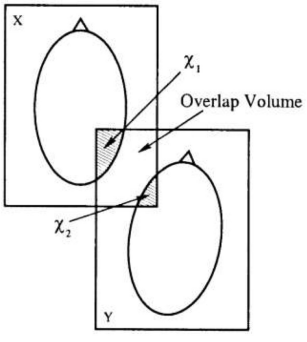

(a) registered (b) mis-aligned by 2 mm (c) mis-aligned by 5 mm



Fuzzy Binning: soft bin boundaries

Treating Partial Overlaps

1. Pad with background value
2. Only compute cost for overlaps
   - → smoothly de-weight close to borders



Optimization Cost Functions

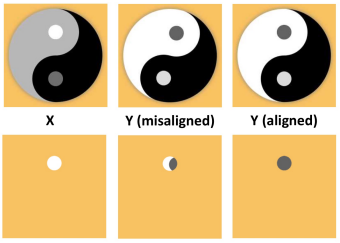
Intra-Modality:

Least-Squares: $C^{LS} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$

Norm. Cross-Correlation: $S^{NC} = \frac{\sum_{i=1}^{N} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{N} {x_i}^2} \cdot \sqrt{\sum_{i=1}^{N} {y_i}^2}}$

Local Cross-Correlation: $S^{LNC} = \frac{\sum_{i=1}^{N} \bar{x}_i \cdot \bar{y}_i}{\sqrt{\sum_{i=1}^{N} {\bar{x}_i}^2} \cdot \sqrt{\sum_{i=1}^{N} {\bar{y}_i}^2}}$

Inter-Modality:



Variance of Intensity Ratios: $C^{VIR} = \sum_{k=1}^{K} \frac{|Y_k|}{N} \frac{\sigma(Y_k)}{\mu(Y_k)}$

→ compute variance within bins

Joint Entropy: $H(X,Y) = -\sum_{i=1}^{K_X}\sum_{j=1}^{K_Y} p_{ij} \log_2 p_{ij}$

→ loss of information when images become more equal
→ minimized through excessive scaling

Mutual Information: $I(X,Y) = H(X) + H(Y) - H(X,Y)$

→ should be maximized

Normalized Mutual Information: $NMI(X,Y) = \frac{H(X) + H(Y)}{H(X,Y)}$

→ works better for varying overlaps

Grid Search

Initialize minimization problem from estimates coming from an evenly-space grid in the search space
→ combinatorial nightmare in higher dimensions

Multi-scale Optimization

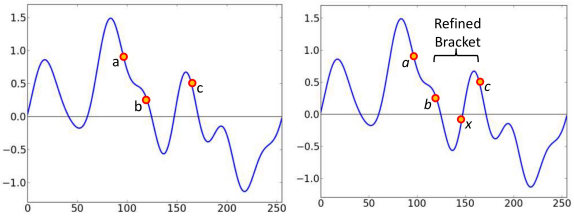
Optimize on lower scale and initialize algorithm on higher scale with that result
→ optimum on different scales must be within same basin

Definition: Simplex

Convex hull of d+1 points in a d-dimensional space

Bracketing in 1D

Take three points (a,b,c):
If: f(a) > f(b) and f(c) > f(b):
there exists a local minimum
Sample new point in between and reset brackets
→ no analogies in higher dimension

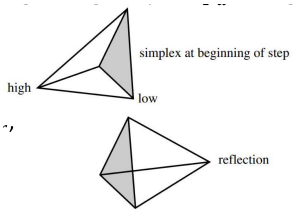


Downhill Simplex (Nelder Mead) Method

0. Initialization: $\mathbf{p}_i = \mathbf{p}_0 + \Delta_i \, \mathbf{e}_i$

Δ_i: dictate the magnitude of the search directions
→ Requires to be non-degenerate
→ initialization of each corner of the simplex

Iterate until convergence:



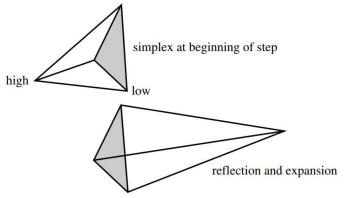
1. Reflection: $\bar{\mathbf{p}} := \frac{1}{d}\sum_{i=0}^{d-1} \mathbf{p}_i$
$\mathbf{p}_r := \mathbf{p}_d + 2(\bar{\mathbf{p}} - \mathbf{p}_d)$

→ reflect point with highest costs through the opposing face

If: p_r is better than the second highest point and worse than the best:

replace p_d with p_r

Else if: p_r is the new lowest point:



2. Expansion: $\mathbf{p}_e := \mathbf{p}_d + 3(\bar{\mathbf{p}} - \mathbf{p}_d)$

→ double expansion in promising search direction

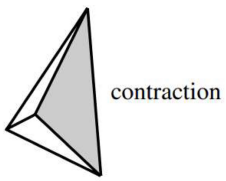
If: $f_e < f_r$: replace p_d with p_e

Else: replace p_r with p_e

Else if: p_r is worse than the second highest point

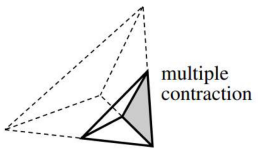
3. Contraction: $\mathbf{p}_c := 0.5(\bar{\mathbf{p}} + \mathbf{p}_d)$

→ move halfway towards opposing face



If: $f_c < f_d$: replace p_d with p_c

If: $f_c \geq f_d$:



4. Multiple Contractions: $\mathbf{p}'_i := 0.5(\mathbf{p}_i + \mathbf{p}_0)$

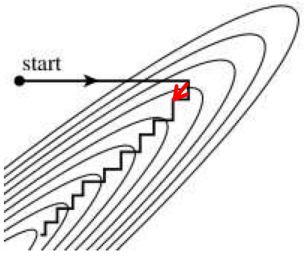
→ contraction in multiple directions leads to more careful steps

Convergence

- absolute difference on costs: $(f_d - f_0)$
- fractional difference on costs: $\frac{2(f_d - f_0)}{|f_d| + |f_0| + \epsilon}$
- maximum distance of points
- maximum iterations
- maximum number of function evaluations

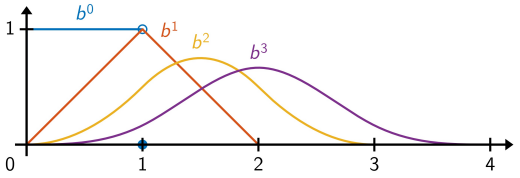
Powell's Method

Problem: Small optimization steps when costs are not aligned with search direction



Approach: After 1D opt. use $p_i - p_{i-d}$ as new search direction

Definition: Cardinal B-Splines
Piecewise Polynomial over compact support

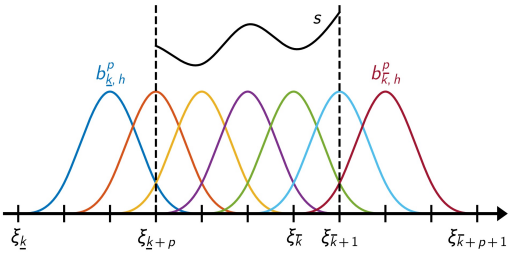


Base Case b^0: Indicator function on the half-open interval $[0,1)$

Recursive Definition: $b^p = b^0 * b^{p-1}$

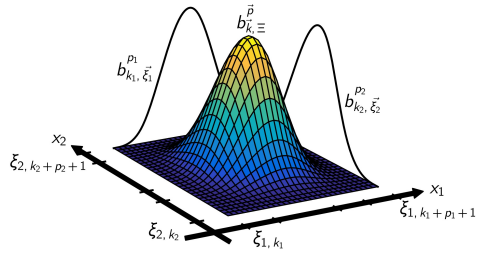
Definition: Uniform B-Splines
Shifted and scaled cardinal B-spline:

$$b^p_{k,h}(x) = b^p\left(\frac{x}{h} - k\right)$$

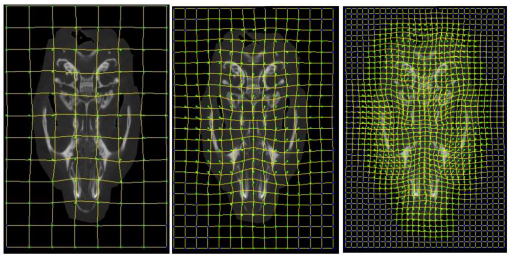


Definition: Multi-dimensional B-Splines
Arise from point-wise product:

$$b^{\mathbf{p}}_{\mathbf{k}}(\mathbf{x}) = \prod_{t=1}^{d} b^{p_t}_{k_t}(x_t)$$



Warp Fields
Represent the transformation through a regular grid on the image where each point is the knot in a multi-dimensional B-spline



Non-linear Optimization Techniques

Gradient Descent: Optimize towards negative gradient
simple: $\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{1}{\lambda}\nabla f(\mathbf{w}_k)$
safe: Perform line search along $-\nabla f(w)$

Newton's Method: Solve for optimum using a second-order Taylor series expansion

Approximation: $f(\mathbf{w}) \approx f(\mathbf{w}_k) + \nabla f(\mathbf{w}_k)(\mathbf{w} - \mathbf{w}_k) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_k)^{\mathrm{T}}\mathrm{H}(\mathbf{w}_k)(\mathbf{w} - \mathbf{w}_k)$

Gradient: $\nabla f(\mathbf{w}) \approx \nabla f(\mathbf{w}_k) + \mathrm{H}(\mathbf{w}_k)(\mathbf{w} - \mathbf{w}_k)$

Update: $\mathbf{w}_{k+1} = \mathbf{w}_k - \mathrm{H}^{-1}(\mathbf{w}_k)\nabla f(\mathbf{w}_k)$

Gauss-Newton: Substitute Newton with Hessian approximation through Jacobins:

Update: $\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathrm{J}_h^{\mathrm{T}}\mathrm{J}_h\right)^{-1}\mathrm{J}_h^{\mathrm{T}}\mathbf{h}(\mathbf{w}_k)$
→ $J^T J$ positive semi-definite, thus more stability
→ easier to compute

Levenberg: Interpolate between gradient descent far from optimum and Gauss-Newton close to optimum

Update: $\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathrm{J}_h^{\mathrm{T}}\mathrm{J}_h + \lambda\mathrm{I}\right)^{-1}\mathrm{J}_h^{\mathrm{T}}\mathbf{h}(\mathbf{w}_k)$
→ dampened Gauss-Newton

Levenberg-Marquardt: Scale dimensions to ensure equal progress along all dimensions

Update: $\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathrm{J}_h^{\mathrm{T}}\mathrm{J}_h + \lambda\mathrm{diag}\left(\mathrm{J}_h^{\mathrm{T}}\mathrm{J}_h\right)\right)^{-1}\mathrm{J}_h^{\mathrm{T}}\mathbf{h}(\mathbf{w}_k)$

Definition: Jacobian Matrix

Jacobian of an N-D vector is a N×N matrix of all first-order partial derivatives

$$\mathbf{v}(x,y) = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} \qquad \mathbf{J}(x,y) = \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix}$$

2D vector field | Jacobian of a 2D vector field

Determinant: Indicator to what happens to local volumes

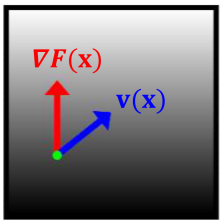
- $|\mathbf{J}|>1$: local expansion
- $|\mathbf{J}|<1$: local compression
- $|\mathbf{J}|=0$: non-invertible
- $|\mathbf{J}|<0$: orientation-reversal

Definition: Optical Flow

Find the displacement $v(x)$ of pixel x between two frames.

Assumption: Intensity of pixel does not change between frames

Approximation: $M(\mathbf{x}) = F(\mathbf{x}) - \langle \mathbf{v}(\mathbf{x}), \nabla F(\mathbf{x}) \rangle$



Aperture problem

We only observe motion perpendicular to above constraint along gradient.

Solution: $\mathbf{v}(\mathbf{x}) = \lambda \nabla F(\mathbf{x})$

$$\mathbf{v}(\mathbf{x}) = \frac{F(\mathbf{x}) - M(\mathbf{x})}{\|\nabla F(\mathbf{x})\|_2^2} \nabla F(\mathbf{x}) \qquad M(\mathbf{x}) = F(\mathbf{x}) - \lambda \|\nabla F(\mathbf{x})\|_2^2$$

$$\lambda = \frac{F(\mathbf{x}) - M(\mathbf{x})}{\|\nabla F(\mathbf{x})\|_2^2}$$

Definition: Diffeomorphism

Transformation with the following properties:

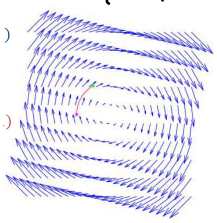
1. Vector field is differentiable
2. Transformation is invertible
3. Inverse is differentiable

Diffeomorphic Registration (LDDMM)

Represent diffeomorphic transformation to velocity field

Properties:

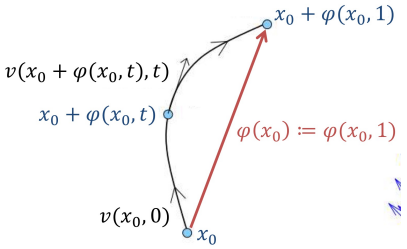
1. Differentiability follows from smoothness of v
2. Differentiability of Inverse



Time-Varying Velocity Fields

Diffeomorphic transformation smoothly varies over time by changing the velocity vector field $v(x,t)$:

$$\frac{\partial \varphi(x,t)}{\partial t} = v(x + \varphi(x,t), t)$$



Limitations:

- Brains might not be diffeomorphic between individuals
- Cannot accurately reproduce manual labels

Regularization of Warp Fields

Problem: B-spline interpolation ensures smooth deformations but still permits drastic deformations such as collapse to a single point

Elastic Registration: seeks a trade-off between image similarity and regularity of the transformation

→ can prevent desired large deformations

Demons Regularization

Idea: Solve optical flow problem and take normal flow, flow with smallest norm, to solve aperture problem and fix numerical issues for small gradients

Optical flow:
$$\mathbf{v}_D(\mathbf{x}) = \frac{F(\mathbf{x}) - M(\mathbf{x})}{\|\nabla F(\mathbf{x})\|_2^2 + (F(\mathbf{x}) - M(\mathbf{x}))^2} \nabla F(\mathbf{x})$$

Observation: keeps correspondences in neighborhood by bounding $\|v(x)\|$

$$\|\mathbf{v}_D(\mathbf{x})\| = \frac{|F(\mathbf{x}) - M(\mathbf{x})| \, \|\nabla F(\mathbf{x})\|}{\|\nabla F(\mathbf{x})\|_2^2 + (F(\mathbf{x}) - M(\mathbf{x}))^2}$$
$$= \frac{\mathrm{GM}^2(|F(\mathbf{x}) - M(\mathbf{x})|, \|\nabla F(\mathbf{x})\|)}{2\,\mathrm{QM}^2(|F(\mathbf{x}) - M(\mathbf{x})|, \|\nabla F(\mathbf{x})\|)} \leq 0.5$$

Algorithm:

Iterate between

1. Compute regularized optical flow $\bar{\psi}(x)$
2. Add optical flow to displacement field: $\hat{\psi}(x) = \psi(x) + \bar{\psi}(x)$
3. Perform Gaussian smoothing: $\varphi(x) = G_s * \hat{\psi}(x)$

PASHA Algorithm

Idea: Optimize energy function with regularization term for image similarity

Energy:
$$E(\varphi, \psi) = \underbrace{\|F(x) - M(\psi(x))\|_2^2}_{\text{L2 Similarity}} + \underbrace{\sigma\|\varphi(x) - \psi(x)\|_2^2}_{\text{Coupling}} + \underbrace{\sigma\lambda\|J_\varphi(x)\|_F^2}_{\text{Regularity}}$$

- ψ: transformation of M
- φ: "correspondences", hidden variables that simplify optimization

Algorithm:

Iterate between:

1. Line search along negative gradient for optimization step:

   Minimize: $\|F(x) - M(\psi(x))\|_2^2 + \sigma\|\varphi(x) - \psi(x)\|_2^2$

   Gradient: $2[F(x) - M(\psi(x))]\,\nabla M(\psi(x)) + 2\sigma[\varphi(x) - \psi(x)]$

2. Regularize the transformation which can be computed through a convolution

   Minimize: $\|\varphi(x) - \psi(x)\|_2^2 + \lambda\|J_\varphi(x)\|_F^2$

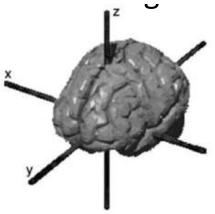
   Convolution: $\varphi = K_\lambda * \psi$

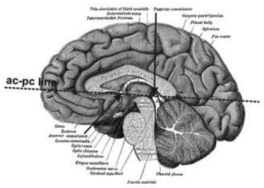
Atlas-Based Labeling

Idea: To label brains register brain against template labeled brain and look up labels for each voxel

Algorithm:
1. Normalize images, apply same transformation to labels
2. In each template voxel, estimate label probabilities
3. Register new brain to template, look up labels

Brain Templates





Talairach Space: Standardized brain coordinate system based on anatomical landmarks:
- Mid-sagittal plane defines yz-plane
- y axis connects anterior and posterior commissure
- Anterior commissure defines origin

→ single brain

MNI Template: Averaged brains aligned to Talairach space:

MNI305: Linearly mapped 305 normal brains to the average from the previous step and averaged them

ICBM152: Linearly mapped 152 normal brains to MNI305; standard used by International Consortium for Brain Mapping (ICBM)

Simple Template Building

Pick representativ individual, register all others to it and average over them

Iterative Template Refinement (SyGN)

0. Initialize template as average ove affine registered images

Iterate until convergence:

1. Register all subjects to template
2. Update appearance of the template
   Minimize the overall dissimilarity between all registered images and template

   Cross-correlation:

   Iterate:
   1. For all images, compute gradients of C^{LNC}
   2. Slightly smooth and average them
   3. Add a fraction of the result to the template

3. Update shape of the template
   Minimize deformation required to register all images
   → factor out mutual part and add to template
   → average respective velocity fields

5. Segmentation

Definition: Segmentation

Partition image into connected regions belonging to same object or material

Definition: Clustering

Identify group of similar points in potential abstract data space

Definition: Classification

Assign voxels or pre-segmented object to a pre-defined category

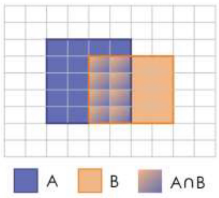
Segmentation in Neuroimaging

1. Brain extraction from MR scans
   → limit region of statistical analysis
2. Volume measurements from brain regions
3. Limit regions for other applications

Evaluation Metrics

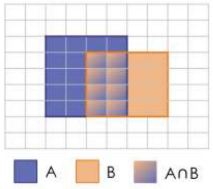
Dice score:

$$DSC(A,B) = \frac{2}{+} = \frac{2\,|A \cap B|}{|A| + |B|}$$



Intersection over union:

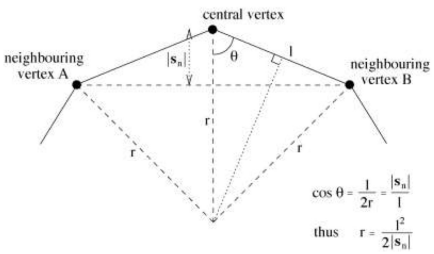
$$IoU(A,B) = \frac{}{+\ -} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} = \frac{|A \cap B|}{|A \cup B|}$$



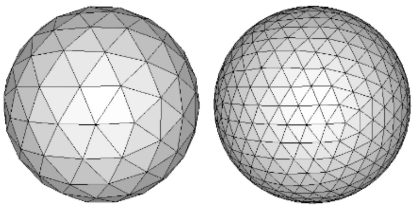
Definition: Local Curvature

Define radius of curvature through neighboring vertices

Curvature: $\kappa = 1/r$

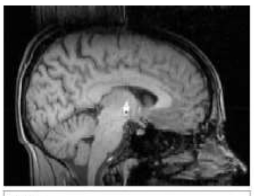


Deformable Contours

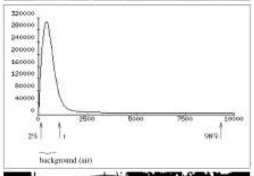


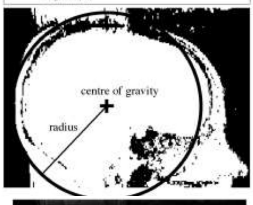
Define a mesh which is fitted iteratively to the brain/region of the brain by moving the mesh's vertices

Initialization:



0. Based on histogram compute bottom and top 2% intensity values: I_2 and I_{98}
1. Compute global intensity threshold:
   $I_{\theta_g} = I_2 + 0.1(I_{98} - I_2)$
2. Compute center of gravity and compute sphere holding the same volume as above-threshold voxels
3. Compute sphere with half of above radius
   → compute vertices from tessalation of sphere





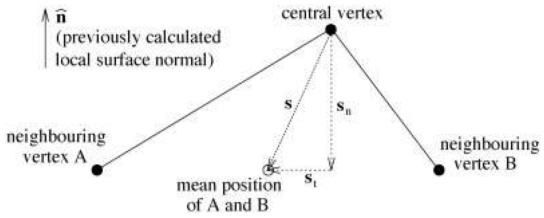
Fitting to Brain Surface

Idea: Compute surface normals and step sizes for each vertex that move vertices towards the brain surface. The inwards or outwards direction is locally decided on an intensity threshold and integrated to the step size

Surface Normals: Estimate through sum over pairwise cross-products of neighboring vertices

Normal component: $\mathbf{s_n} = (\mathbf{s} \cdot \mathbf{n})\mathbf{n}$

Tangential components: $\mathbf{s_t} = \mathbf{s} - \mathbf{s_n}$

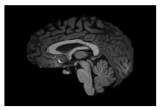


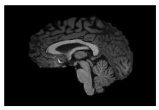
Imin: $I_{\min} = \max\left(I_2, \min\left(I_m, I(0), I(1), \ldots, I(d_1)\right)\right)$

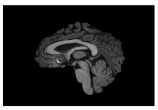
Imax: $I_{\max} = \min\left(I_m, \max\left(I_{\theta_g}, I(0), I(1), \ldots, I(d_2)\right)\right)$

→ found via inward search along inverse normal for a certain distance

Local Threshold: $I_{\theta_l} = I_2 + b(I_{\max} - I_2)$


b=0.25


b=0.5


b=0.75

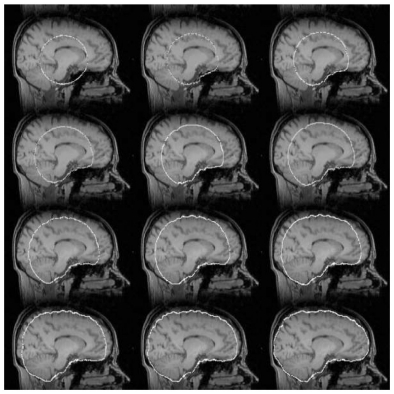
Maximum speed: $f_D = \dfrac{2(I_{\min} - I_{\theta_l})}{I_{\max} - I_2}$

Outward motion ($f_D > 0$) if $I_{\min} > I_{\theta_l}$

Inward motion ($f_D < 0$) if $I_{\min} < I_{\theta_l}$

Adaptive Speed: $f_S = \dfrac{1 + \tanh\left(\left(\frac{1}{r} - E\right)F\right)}{2}$ $\quad E = (1/r_{\min} + 1/r_{\max})/2$ $\quad F = 6/(1/r_{\min} - 1/r_{\max})$

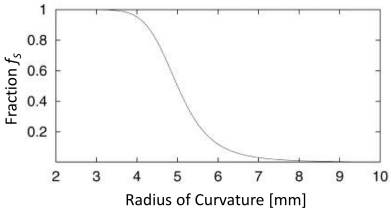
→ regularized by local curvature



Observation:

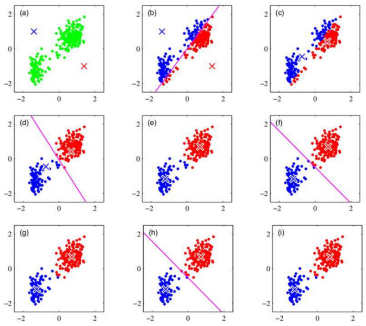
f_S should be
- Close to zero for small curvature $\kappa < 1/r_{\max}$
- Close to one for large curvature $\kappa > 1/r_{\min}$

Make use of $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$



k-Means Segmentation

Idea: Represent each pixel in a feature space and perform k-means clustering in it



Objective:

Minimize: $D = \sum_i \|\mathbf{v}_i - \boldsymbol{\mu}_{\gamma(i)}\|^2$

→ we want to minimize the overall distance to the assigned cluster center of each point

→ no closed-form efficient solution

Proof: If μ_k are fixed we minimize objective with:

$$2\sum_{i \in C_j}(\mathbf{v}_i - \boldsymbol{\mu}_j) = 0 \;\Leftrightarrow\; \boldsymbol{\mu}_j = \frac{\sum_{i \in C_j} \mathbf{v}_i}{|C_j|}$$

Algorithm: v_i: data points, μ_k: centers, $\gamma(i)$: assignment

0. Initialize cluster centers μ_k
   → random, subset of data, grid, prior knowledge

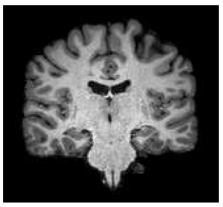
Iterate until convergence:

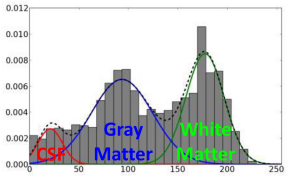
1. Compute assignments $\gamma(i)$
   → assign to closest center
2. Compute centers μ_k
   → center of gravity over assigned points

Problems:

- converges to local minima
- sensitive to initial clusters
- spherical clusters of similar size
- fixed number of clusters

Gaussian Mixture Models





Idea: Represent data through a mixture of gaussians where each gaussian corresponding to a semantic meaning

Definition: $p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k^2)$ $\quad N(x|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$

$\sum_{k=1}^{K} \pi_k = 1$

μ_k: center

σ_k^2: variance

π_k: mixing coefficients $\pi_k = \frac{1}{n}\sum_{i=1}^{n} z_{ik}$

z_i: membership vector for each sample $\mathbf{z}_i \in \{0,1\}^K$

Expectation Maximization

E-Step: Compute cluster probability

$$\rho_{ik} = p(z_{ik} = 1|x_i) = \frac{p(x_i|z_{ik}=1)\, p(z_{ik}=1)}{p(x_i)} = \frac{N(x_i|\mu_k, \sigma_k^2)\pi_k}{\sum_{l=1}^{K} \pi_l N(x_i|\mu_l, \sigma_l^2)}$$

M-step: Compute GMM parameters μ_k, σ_k^2, π_k

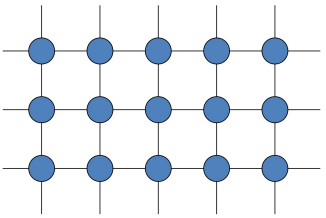
Maximize: $p(X|\pi,\mu,\sigma) = \prod_{i=1}^{n}\sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k^2) \Leftrightarrow \ln p(X|\pi,\mu,\sigma) = \sum_{i=1}^{n} \ln \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k^2)$

Parameters: $\mu_k = \frac{\sum_{i=1}^{n} \rho_{ik} x_i}{N_k}$; $\sigma_k^2 = \frac{\sum_{i=1}^{n} \rho_{ik}(x_i - \mu_k)^2}{N_k}$; $\pi_k = \frac{N_k}{n}$

Problems:

- slower than k-Means
- converges to local optimum
- components can collapse to a single point

Markov Random Fields



Idea: To model spatial dependencies define neighborhood graph over image and define prior on label through the neighborhood pixels

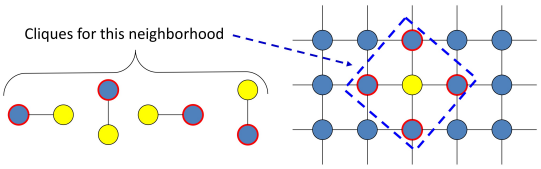
$$p(z_S|x_S) = \frac{p(x_S|z_S)\,p(z_S)}{p(x_S)}$$

Theorem: Hammersley-Clifford

If density of z_S is positive, it factorizes over the cliques C of a MRF:

$$p(z_s) \propto \prod_C p(C) = e^{-\sum_C V_c(z_C)}$$

V_c: clique potentials



Generalized Potts Model:

→ widely used potential for cliques

$$V_{ij}(z_i, z_j) = u_{ij}\delta(z_i \neq z_j)$$

u_{ij} = Penalty for discontinuity between voxels *i* and *j*; in the simplest case, the same number β for all pairs

$\delta(z_i \neq z_j) = 1$ if $z_i \neq z_j$, 0 else

Unary Potentials: $V_i(x_i, z_i) = \ln(\sqrt{2\pi}\sigma_{z_i}) + \frac{(x_i - \mu_{z_i})^2}{2\sigma_{z_i}^2}$

→ modelled through gaussian model $p(x_i|z_i) = \frac{1}{\sqrt{2\pi}\sigma_{z_i}} e^{\frac{(x_i-\mu_{z_i})^2}{2\sigma_{z_i}^2}}$

MAP Estimation: Compute assignments maximizing $p(x_i|z_x)$

Minimize energy over potentials: $\sum_i V_i(x_i, z_i) + \sum_i \sum_{j \in NB(i)} V_{ij}(z_i, z_j)$

Data Term External Energy — Smoothness Term Internal Energy

HMRF-EM Algorithm

E-Step: Compute clique assignments

1. Compute MRF-MAP estimate (hard labels)
2. Calculate posterior distribution of labels:

$$\rho_{ik} = p(z_{ik} = 1|x_i) = \frac{p(x_i|z_{ik} = 1)\,p(z_{ik} = 1|N(i))}{p(x_i)}$$

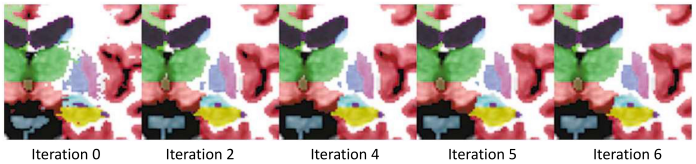
M-step: Estimate gaussian parameters for unary potentials

→ use standard EM definition

Iterated Conditional Modes

Heuristic approach to MAP estimation

1. Start with a reasonable initialization (e.g., without pairwise potentials)
2. Iterate over all voxels:
   - Condition on all the neighbors (treat them as fixed)
   - Pick the label that minimizes the energy
3. Repeat step 2 until convergence

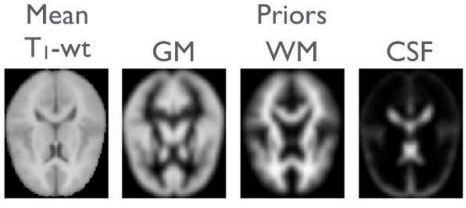


Priors from Atlas

Take tissue class probabilities from an atlas as prior

→ can help with difficult cases

→ requires registration



Freesurfer

MR image-processing library with MRF support

Non-stationary: linear transformation to an atlas strongly reduces number of possible labels

Anistropic: Pairwise potentials capture directional relationship between labels

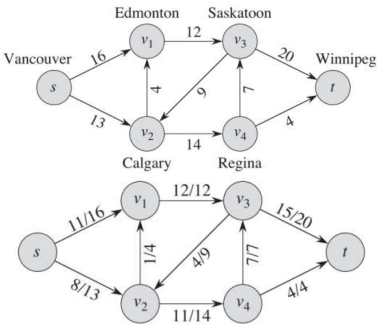
Graph Cuts

Flow network:

Graph: $G = (V, E)$

Edge capacities: $c(u, v) \geq 0$

→ edges with limited capacity



Flow: Real-valued function $f: V \times V \to \mathbb{R}$, such that:

Capacity constraint: $0 \leq f(u, v) \leq c(u, v) \; \forall u, v$

flow conservation: $\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) \; \forall u \in V - \{s, t\}$

Max-Flow problem: Maximize flow through flow network

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

Cut: Partition of graph into two cliques **cut** (S, T)

Capacity of Cut: $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$

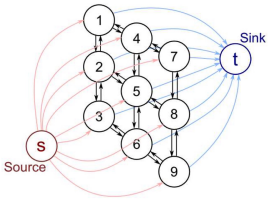
Net-Flow across Cut: $f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$

Min-Cut: Minimal costs along cut

Max-Flow/Min-Cut Dualism: Minimum cut equals to edges with saturated capacities

Lemma: For any cut(S,t) and flow f:

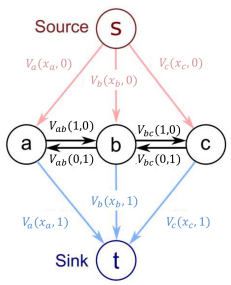
$$|f| = f(S, T) \leq c(S, T)$$



Binary Labels:

Unary costs: edges from source and to sink

Pairwise costs: capacities between pixels



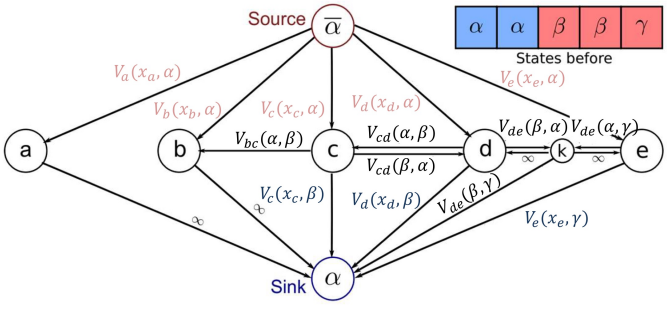
Multi Label (Alpha Expansion)

Iteratively expand labels with separate graph

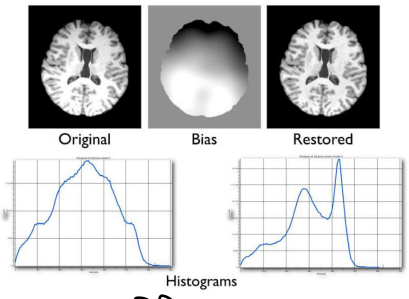
Potentials have to form metric:

$$V_{ij}(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$$
$$V_{ij}(\alpha, \beta) = V_{ij}(\beta, \alpha) \geq 0$$
$$V_{ij}(\alpha, \beta) \leq V_{ij}(\alpha, \gamma) + V_{ij}(\gamma, \beta)$$

Graph Construction:



Eliminating Bias Fields



Bias Field: Inhomogeneities in the magnetic field cause a bias in specific regions

Model: inhomogeneities are multiplicative and smoothly varying
→ use log-intensities to turn bias in linear offset

Likelihood: $p(y_i|z_{ik}=1,b_i)=\frac{1}{\sqrt{2\pi}\sigma_k}e^{-\frac{(y_i-\mu_k-b_i)^2}{2\sigma_k^2}}$

Prior: $p(b_S)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_b|^{\frac{1}{2}}}e^{-\frac{1}{2}b_S^T\Sigma_b^{-1}b_S}$

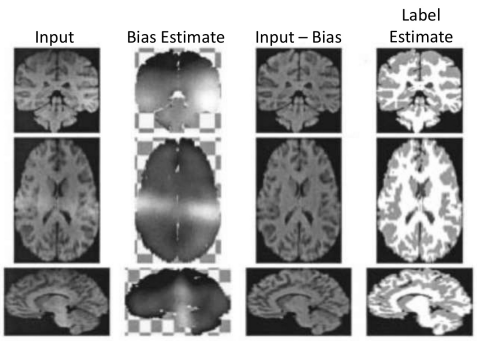
Estimation:
Iteratively update bias field according to map estimation:

Update: $b_S = Hr_S$ with $H=\left[\Sigma_m^{-1}+\Sigma_b^{-1}\right]^{-1}$
→ H might be infeasible to compute
→ H should act as low-pass filter

Approximation: $b_S=\frac{Fr_S}{F\Sigma_m^{-1}\mathbf{1}}$ ← component-wise division

Derivation:
1. Maximize posterior $\nabla_{b_S}(\ln p(y_S|z_S,b_S)+\ln p(b_S))=\mathbf{0}$
2. Compute likelihood: $\frac{\partial}{\partial b_i}\ln p(y_S|z_S,b_S)=\sum_{k=1}^{K}\rho_{ik}\frac{y_i-\mu_k-b_i}{\sigma_k^2}=\underbrace{\sum_{k=1}^{K}\rho_{ik}\frac{y_i-\mu_k}{\sigma_k^2}}_{\text{Define a "residual" vector } r_S \text{ whose } i\text{th coefficient this is}}-\underbrace{b_i\sum_{k=1}^{K}\frac{\rho_{ik}}{\sigma_k^2}}_{\text{Define a diagonal "inverse covariance" } \Sigma_m^{-1} \text{ whose } i\text{th element this is}}$
3. Insert vectorized form of likelihood definition:
$r_S-\Sigma_m^{-1}b_S+\nabla_{b_S}\ln p(b_S)=\mathbf{0}$
4. Compute gradient of prior: $\nabla_{b_S}\ln p(b_S)=-\Sigma_b^{-1}b_S$
5. Final form follows directly: $b_S=Hr_S$ with $H=\left[\Sigma_m^{-1}+\Sigma_b^{-1}\right]^{-1}$

HMRF-EM with Bias Fields



E-step:
1. **New: Estimate bias field (according to rule above)**
2. Compute MRF-MAP estimate (hard labels)
   - **New: Condition on bias field**
3. Calculate posterior distribution of labels:

$$\rho_{ik}=p(z_{ik}=1|y_i,\boldsymbol{b_i})=\frac{p(y_i|z_{ik}=1,\boldsymbol{b_i})\,p(z_{ik}=1|N(i))}{p(y_i)}$$

M-step:
- **Account for bias** when estimating μ_k and σ_k^2

Modification for Non-Tissue Classes

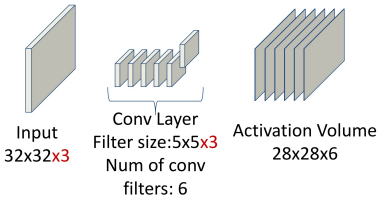
Problem: Non-tissue classes such as CSF or air are poorly described by gaussians

Solution: Model these classes with a uniform distribution

Convolutional Neural Networks

Perform segmentation hierarchical through a number of convolutions
- → pre-trained models allowing zero-shot inference or fine-tuning
- → no initial registration required
- → represent complex spatial correlations

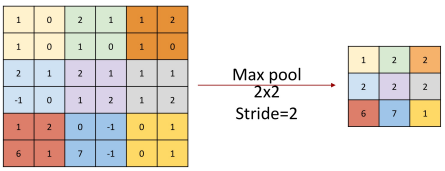
Convolution Layer: Compute convolution with kernel over multiple channels



Rectified Linear Unit: $f(\alpha) = \max(\alpha, 0)$

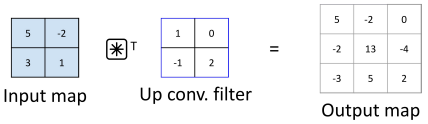
Parametric: $f(\alpha) = \max(m\alpha, \alpha)$ m: slope

Pooling: Pool local features into feature map of smaller resolution
- → max, average or stochastic



Transposed convolution:

Perform a convolution that reverses the dimensionality reduction of a standard convolution. This is done by inserting (stride-1) rows and columns between each row and column



Softmax: $\frac{e^{s_k}}{\sum_j e^{s_j}}$

Gray Matter	3.0	20.08	0.87
White Matter	1.0	2.7	0.12
CSF	-2.5	0.08	0.00

exp → normalize

→ used to convert logits into a probabilistic prediction

Losses:

Cross-entropy: $L_i = -ln\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$

→ minimize the NLL for the prediction of the class

Weighted Loss: $w(x) = w_c(x) + w_0 \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right)$

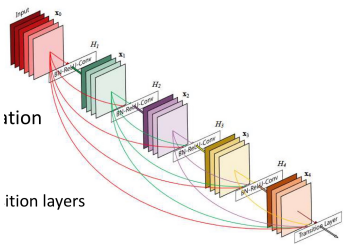
→ used to accentuate boundaries & improve minority classes

Dice Loss: $\mathrm{DL}(a, b) = 1 - \frac{2\sum_i a_i b_i + \epsilon}{\sum_i a_i^2 + \sum_i b_i^2 + \epsilon}$

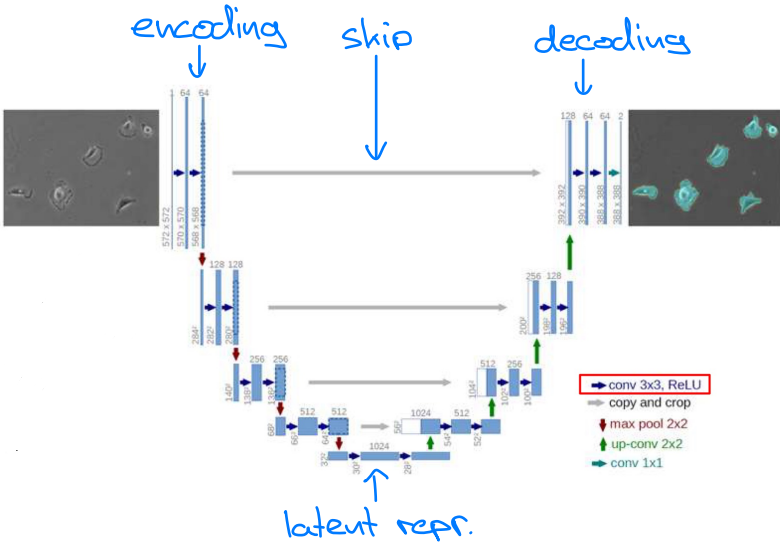
Augmentations: Apply random transforms to increase amount of data and/or make task harder for improved performance

Batch Normalization: Learnable normalization of the input across batch
- → improves stability and speeds up training

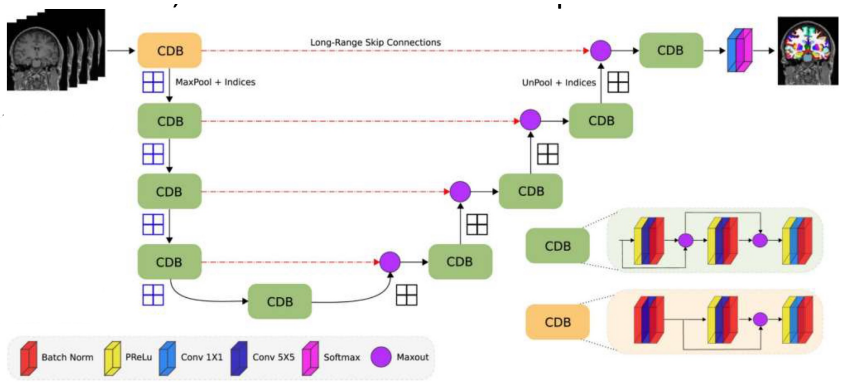
DenseNets: Connect layer to all its predecessors to encourage extraction of new features instead of preservation of old ones



U-Net: Encode image into latent representation and decode into segmentation map using intermediate features through skip connections through the encoding stage

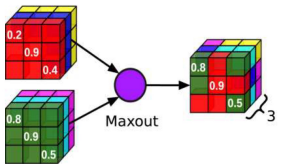


FastSurfer: U-Net with improvements: competitive dense blocks, batch norm., param. ReLU, unpooling
→provides pipeline the inner and outer cortical surfaces



Competitative Dense Blocks

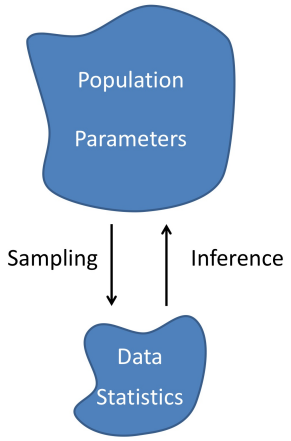
Residual connection from first layer and point-wise max operation introduces competition between features



Strategies:

1. **Using 3D context:** Provide 3 additional slices on both sides, only segment the central one
2. **View aggregation:** Segment after slicing along all three axes. Obtain final segmentation as weighted average of the three results

6. Statistical Testing



Hypothesis Testing

Reject a hypothesis about the population parameters by showing that it is highly unlikely given the observed data.

Population: Set about we want to make statement
- → typically not observable
- → only to a certain degree samplable

Sample: set of individuals able to observe

Null hypothesis: hypothesis we want to reject H_0
- → inverse to the research hypothesis

Errors:

Type I: reject true null hypothesis
- → controllable by design of statistical test
- → typical acceptance level: $\alpha = 5\%$

Type II: fail to reject false null hypothesis

H_0	True	False
Reject	Type I	Correct
Reject	Correct	Type II

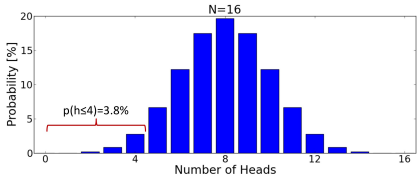
Problems:
- ignores prior probabilities
- hypothesis might be rejected due to unmodeled variation in the sample
- Rejected null hypothesis does not imply unambiguous causal dependence
- Publication bias: only positive results are published

Definition: Test Statistic

Statistical quantity derived from data and used as basis for the test.

Definition: P-Value

Conditional probability to observe a computed value or a more extreme one of a test statistic if H_0 is true
- → smaller p: stronger evidence against H_0 ($p < 0.05$)



One-sided: compute p-value on one tail of the distribution

Two-sided: Compute two one-sided tests on both tails
- → suitable if no prior knowledge of bias in any direction

Caveats:
1. Not a probability of committing type I error
2. No direct measure of effect size
3. Not the probability that null hypothesis is true

t-Tests

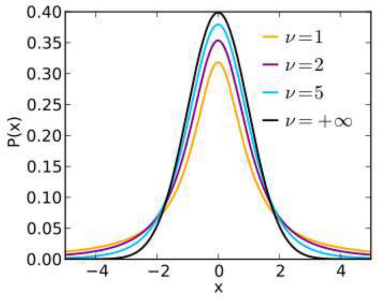
Idea: t-Tests are appropriate to evaluate whether random variables are distributed with a given mean. For this we check how many sample standard deviations mean and tes

Example: Do chess players have larger brains on average?

Hypotheses: H_0: μ=1260 (from literature); H_A: μ>1260 (one-sided)

Sample:

1. Measured brain volumes of n=6 male chess players:
$$v_i = \{1400, 1220, 1280, 1360, 1290, 1350\}$$
2. $v_i^{NC} = \{1190, 1210, 1310, 1370, 1250, 1230\}$



Student t-Distribution

Since we take a sample variance t follows not a normal distribution but a student-t distribution with less degrees of freedom

Single-Sample t-Test

Try to reject the null hypothesis that our samples were drawn from a distribution with pre-defined mean.

Algorithm:

1. Compute deviation of sample mean from H_0:
$$\bar{x} = \frac{\sum_{i=1}^{n} v_i}{n} = 1316.7;\ \bar{x} - \mu_0 = 56.7$$
2. Compute sample variance: "degrees of freedom"
$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(v_i - \bar{x})^2 = 4266.7;\ s = 65.3$$
3. Compute standard error of sample mean:
$$\frac{s}{\sqrt{n}} = 26.7$$
4. Compute *t* score, our test statistic:
$$t = \frac{\bar{x} - \mu_0}{s}\sqrt{n} = 2.13$$

In our example, the one-sided *p* value for $t = 2.13$ and $\nu = 5$ is $p = 0.043$
- Since $p < 0.05$, we reject H_0
- We conclude that chess players have larger brains

Studen-t distribution: $V = n-1$ DoF

Two-sample t-Test

Take second sample to test for the librature value and compare variance of two samples

Algorithm:

1. Compute difference of sample means:
$$\overline{x^{NC}} = 1260;\ \overline{x^{C}} - \overline{x^{NC}} = 56.7$$
2. Compute pooled sample variance (assuming same variances):
$$s_{NC}^2 = 4600;\ s_{C,NC}^2 = \frac{(n_C-1)s_C^2 + (n_{NC}-1)s_{NC}^2}{n_C + n_{NC} - 2} = 4433.3;\ s_{C,NC} = 66.6$$
3. Compute standard error of *difference of* sample means:
$$s_{C,NC}\sqrt{\frac{n_C + n_{NC}}{n_C \cdot n_{NC}}} = 38.4$$
4. Compute t score, our test statistic:
$$t = \frac{\overline{x^C} - \overline{x^{NC}}}{s_{C,NC}}\sqrt{\frac{n_C \cdot n_{NC}}{n_C + n_{NC}}} = 1.47$$

Again, t follows Student's t distribution
- In this case, $\nu = n_C + n_{NC} - 2$
- One-sided *p* for t=1.47 and ν=10 is *p*=0.086
- Test fails to reject H_0! **Why?**
  - We are now considering the difference of sample means, a random variable with larger variance.

Student-t distribution: $V = n_C + n_{NC} - 2$

Welch's modification:
$$t = \frac{\overline{x^C} - \overline{x^{NC}}}{\sqrt{\frac{s_C^2}{n_C} + \frac{s_{NC}^2}{n_{NC}}}}$$

→ apply when variance of the two populations are not equal

Paired t-Test

Standard t-test assumes independent measurements which is often violated in coupled tests, repeated measurements etc..

For pairwise associations (e.g. before/after), test difference between corresponding values through single-sample t-test.

Family-Wise Errors

If we perform multiple test using a test-wise significance α the overall probability of making a type I error is still high Thus we have to take the number of comparisons into account.

→ severe in neuroimaging where we have large statistical maps/volumes of brains

Family-Wise Error Correction

Ad-hoc Approaches: adapt the threshold until map looks right

Sidak correction: Demand a stricter α_{IND} dependent on the number of comparisons

$$1-(1-\alpha_{IND})^N = \alpha_{FW} \text{ iff } \alpha_{IND} = 1-\sqrt[N]{(1-\alpha_{FW})}$$

Bonferroni correction: Computational more convenient approximation of above

$$\alpha_{IND}=\alpha_{FW}/N$$

→ too conservative for correlated neighboring pixels

→ α can become too small for large N

Random Field Theory

Study of the properties of a Gaussian random fields $Z(x)$ which provides theoretical results for smooth statistical maps.

Gaussian Random Fields:

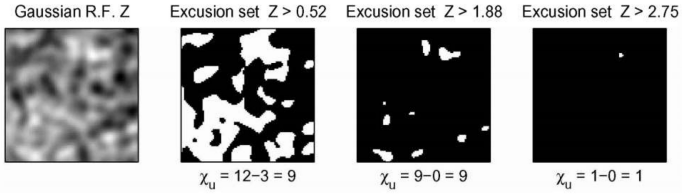
Models statistical processes where each pixel is drawn from a gaussian distribution with a correlation function modeling the correlation bet. pixels

Null hypothesis: $Z(x)$ is gaussian distributed with zero mean and unit variance

Gaussianize RV: $Z = \mathrm{CDF}_G^{-1}(\mathrm{CDF}_t(t))$

Excursion Set: $\{\mathbf{x} \in \Omega \,|Z(\mathbf{x}) > u\}$

→ thresholded set of pixels from the GRF



Euler Characteristic: $\chi_u = \#\text{components} - \#\text{holes}$ of excursion set

→ indicator function if FWE are present

$$p_{FWE}(u) = P\left(\bigcup_i Z_i \geq u\right) = P\left(\max_i Z_i \geq u\right) \approx P(\chi_u > 0) \approx E[\chi_u]$$

Probability of FWE

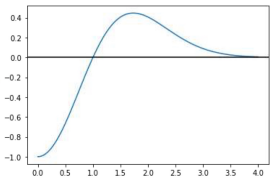
Assumption: $Z(x)$ is defined on 3D volume V and spatially smooth

For sufficiently large u:

Λ: covariance matrix of gradient ∇Z

$|\Lambda|$: indicates roughness

$$p_{FWE}(u) \approx E[\chi_u] \approx \frac{V\sqrt{|\Lambda|}}{(2\pi)^2} e^{-\frac{u^2}{2}}(u^2-1)$$



Alternative: $$p_{FWE}(u) \approx R \frac{(4\ln 2)^{\frac{3}{2}}}{(2\pi)^2} e^{-\frac{u^2}{2}}(u^2-1)$$

Resolution element: $$R = \frac{V}{FWHM_x \times FWHM_y \times FWHM_z}$$

Full-width Half-maximum: width of hypothetical gaussian when at half of its mean's probability

Cluster-level Testing

Idea: Build clusters through thresholding the statistical map and test between clusters. This filters out false positives as true effects typically happen over neighboring regions

Approach:

1. Form clusters by heuristically thresholding statistical maps at some level (e.g., *t*>3)
2. Formal testing can be done **cluster-wise**
   - Size (number of voxels) or mass (sum of scores) serve as test statistics
   - **But:** How to determine null distributions?

→ RFT can be used to define null distribution
→ heavy tails in real-world distribution violates RFT assumptions

Permutation-Based Testing

Idea: Use the permutations of the groups to compare our test statistic against. Ideally this should not change the outcome but correlations between samples might interfere

Null hypothesis: The assignment to the individual groups is unrelated to the test statistic

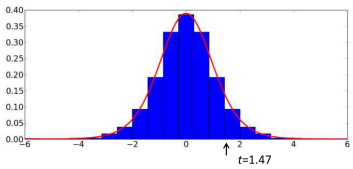
Required number of permutations:

$$p \pm 2\sqrt{p(1-p)/n}$$

Re-consider example of chess-player brain size:

- $v_i^C = \{1400, 1220, 1280, 1360, 1290, 1350\}$
- $v_i^{NC} = \{1190, 1210, 1310, 1370, 1250, 1230\}$

The 12 measurements can be labelled in $\binom{12}{6} = 924$ ways, leading to the following distribution of *t* scores:



Red: Student's t-Distribution
p=0.086 (one-sided)

Blue: Permutation-based Distribution
p=0.091

Cluster-/Permutation-based Correction

For each permutation of group labels:

– Compute per-voxel test statistic (e.g., t-Test)
– Threshold map at some level, compute clusters
– Store the largest / heaviest cluster in the full brain
  - This is the relevant cluster to control FWE rate!

For each cluster found using the true labels:

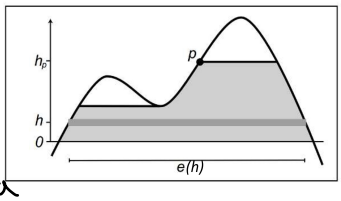
– Compute per-cluster "FWE-corrected" *p* value by comparing size / mass to the null distribution

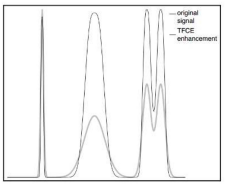
Threshold-Free Cluster Enhancement

Traditional cluster-based analysis requires a threshold that heavily influences the results

Idea: Boost values with large neighboring values to enhance clusters and perform voxel-based analysis with the benefits of cluster-based analysis

Transformation: $TFCE(p) = \int_{h=h_0}^{h_p} e(h)^E h^H dh$





Properties:
- only enhances positive values
- preserves locations of maxima

Permutation-based:
- Compute per-voxel test statistic (e.g., t-Test)
- Apply TFCE to the map of t values
- Store the largest value in the full brain
- Testing compares per-voxel TFCE value to the null distribution

Analysis of Variance (ANOVA)

Idea: Compare the means of different groups by partitioning the overall variance σ^2 into a between-group variance σ^2_{BG} which indicates a difference and a within-group variance σ^2_{WG} which accounts for random effects. The ratio $\sigma^2_{BG}/\sigma^2_{WG}$ is the test statistic indicating differences.

→ with two groups of equal variance equal to two-sample t-test
→ two sided as we test for $\mu_1 \neq \mu_2$
→ one-tailed as we only test on one end of distribution

Generative interpretation: $y_{ij} = \underbrace{\mu + a_i}_{=\mu_i} + \epsilon_{ij}$ with $\sum_j \epsilon_{ij} = 0 \; \forall i$ and $\sum_i N_i a_i = 0$

→ data is generated through effect a_i of a group giving the offset from the grand mean μ and a random effect

Example:

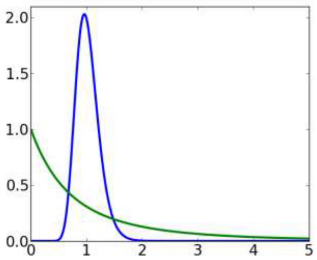
Null hypothesis: Mean intensity does not depend on the task

Data: Intensities of visual cortex during diff. tasks

Task	Measurements		
1: Rest	1	2	2
2: Checkerboard	5	6	5
3: Beep	2	1	

Algorithm:

1. Compute sample estimates of group means μ_i and grand mean μ:
   $\overline{x_1} = 1.67; \overline{x_2} = 5.33; \overline{x_3} = 1.5; \bar{x} = 3$
2. Compute between-group sum of squares:
   $$SS_{BG} = \sum_{i=1}^{M} N_i(\overline{x_i} - \bar{x})^2 = 26.17$$
   – Degrees of freedom: *M-1=2*
3. Compute within-group sum of squares:
   $$SS_{WG} = \sum_{i=1}^{M} \sum_{j=1}^{N_i} (y_{ij} - \overline{x_i})^2 = 1.83$$
   – Degrees of freedom: *N-M=5*
4. Compute the ratio of both, divided by their respective degrees of freedom, as our test statistic:
   $$F = \frac{SS_{BG}/(M-1)}{SS_{WG}/(N-M)} = 35.68$$



Blue: d1=d2=100
Green: d1=2, d2=5

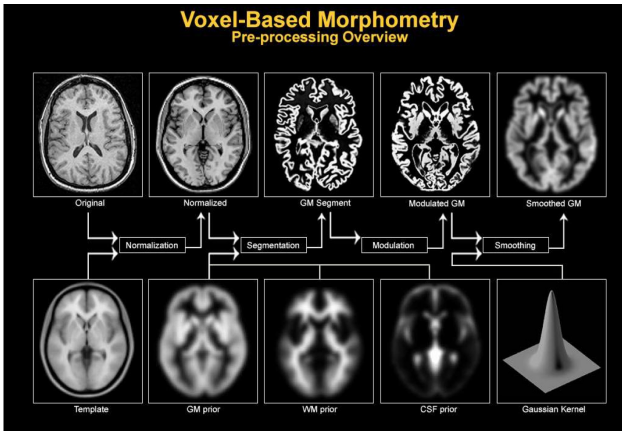
→ F follows a F distribution with parameters d_1 and d_2 which describe the DoF in nominator and denominator

Two-way: Model different factors in different dimensions each with multiple levels. Each combination is entered as a cell an acts like a own group as before.
→ assumes balanced number of samples

$$y_{ijk} = \underbrace{\mu + a_i + b_j + (ab)_{ij}}_{=\mu_{ij}} + \epsilon_{ijk}$$

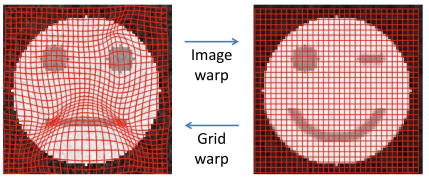
Voxel-Based Morphometry (VBM)

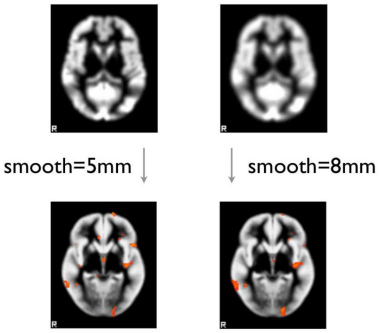
Tool to study gray or white matter volumes in specific brain regions.



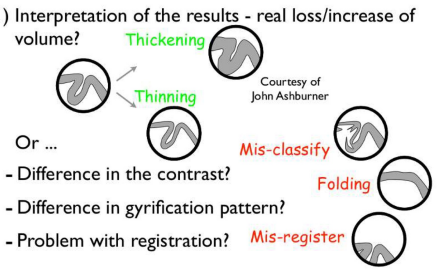
Processing steps:

1. Normalization
   - → register brain to a template through rigid, then affine and then non-linear registration
2. Segmentation
   - → tissue classification in GM/WM/CSF
   - → MRF to deal with noise and bias
3. Modulation
   - → non-linear warping distorts volume
   - → correction by multiplying the normalized gray matter map with the Jacobian determinant of the deformation field
4. Smoothing
   - → apply gaussian smoothing to comensate for inaccurities in normalization



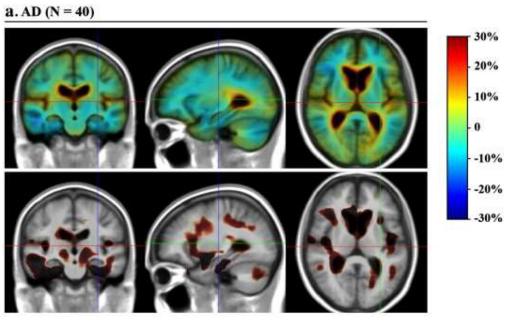


Problems:

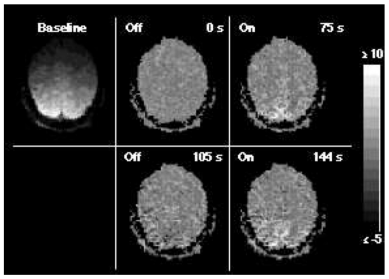


Tensor-Based Morphometry

Concerned with the analysis of the deformation field



7. Functional MRI



Functional MRI

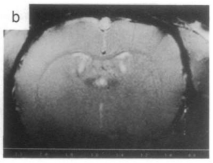
Take repeated measurements of brain after a stimulus to measure function of the brain over time and map functions of brain regions

BOLD Effect

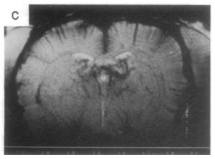
the MR intensity is blood oxygenation level dependent

Oxygenated blood: weakly counteracts the local magnetic field
→ almost no change in signal

Deoxygenated blood: slightly enhances the field
→ decreases T_2^* time and is visible in scan
→ measured in functional MRI

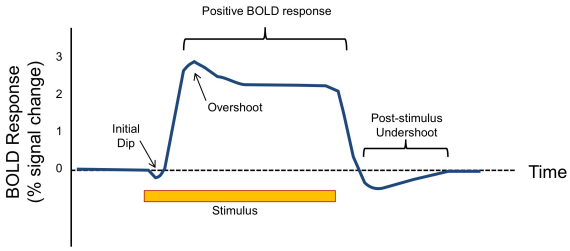


Rat breathing 3% halothane in O_2



Rat breathing 100% N_2

Response to stimulus:



Initial dip: neuronal activity uses up oxygen

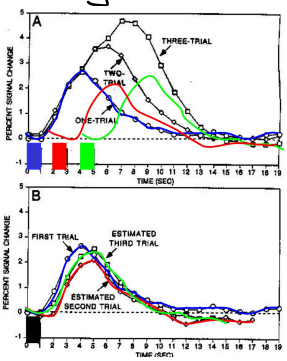
Rise: vescular system provides more oxygenated blood
→ fast and typically not detected
→ overshoots due to preventionary additional blood

Plateau: During neural activity oxygen value is overcompensated

Post-stimulus: undershoot due to more oxygen intake than expected

Problem: BOLD response also visible in larger veins draining blood

Hemodynamic Response Function



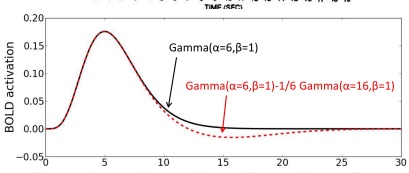
Describes the MR signal in response to a short stimulus
→ variable dependent on subject and regions

Linearity: in most cases we assume that the HRF is approximately linear

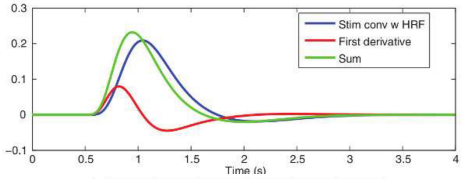
Caveats:

Spacing: responses to stimuli less than two seconds are slightly smaller

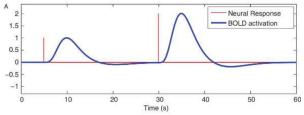
Duration: very short stimuli have a much larger response than expected

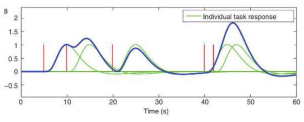


Canonical HRF: Modeling via the Gamma dist.
→ double gamma includes undershoot



Derivative HRF: $f(t_1) \approx f(t) + f'(t)(t_1 - t)$
$f(t+\delta) \approx f(t) + \delta f'(t)$
→ allows better flexibility to adapt to arbitrary HRF





BOLD signal: convolution of the neural response f with HRF h

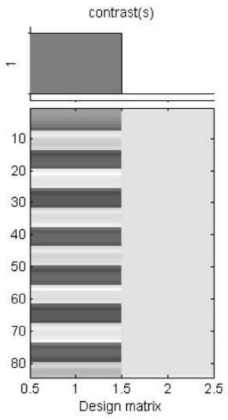
General Linear Model

Provides a unified framework for statistical tests. Tests such as t-tests, F-tests, ANOVA etc. can be represented through a GLM through specific construction of the design matrix
→ widely used in neuroimaging
→ can regress out nuisances by adding a parameter and column in the design matrix which captures unwanted correlations

Model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ — y: measurements; X: design matrix → defines test; β: parameters of test

Null hypothesis: β_0 can be written in general form: $\mathbf{c}^T\boldsymbol{\beta} = 0$ with contrast vector c

Test statistic: $t = \frac{\mathbf{c}^T\hat{\boldsymbol{\beta}}}{\sqrt{\hat{\sigma}^2\mathbf{c}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{c}}}$ with $\hat{\sigma}^2 = \frac{\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}}{N-(p+1)}$



Multivariate Model: $\mathbf{y} = \beta_0 + \beta_1\mathbf{x}_1 + \cdots + \beta_p\mathbf{x}_p + \boldsymbol{\epsilon}$

Hypothesis testing in fMRI

1. Compute the expected BOLD timecourse by convolving the stimulus with HRF
2. Use this as the column of design matrix
   → expected measurement
3. Null hypothesis corresponds to $\beta_i = 0$
   → no response to the stimulus

fMRI Group Analysis

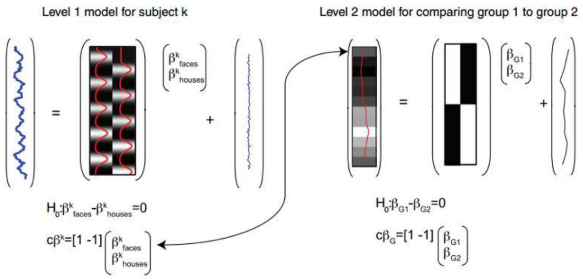
Combines data from multiple subjects to make hypotheses about the population and compensate for variances and DoF over time and between subjects
→ simply normalizing and temporally concatenating scans disregards those effects ("fixed effects analysis")

Approach: Two-Level Analysis

Random effects are estimated in two steps using GLMs:

1. Estimate activation per-subject
2. Test contrasts against zero or compare contrasts between groups

→ weighted least-squares fit in second step can account for within-subject variances



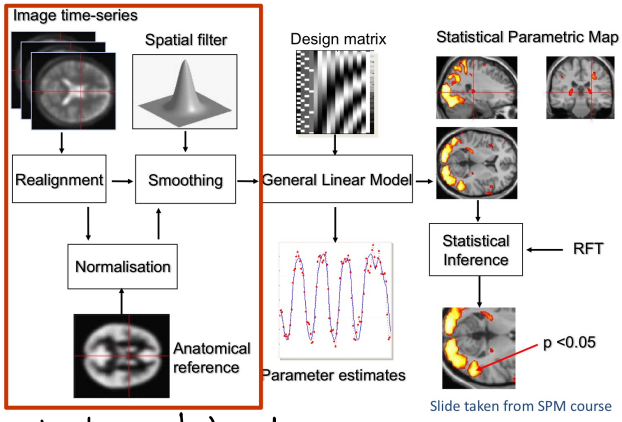
fMRI Pitfalls

Dead Salmon: dead fish showed neural response to stimuli
→ there are always false positives and have to be interpreted carefully

Cluster failure: invalid gaussian assumption on autocorrelation leads to much higher rates of false positives in real world

Correlation in regions of interest: Computing the correlation in pre-selected areas dependent on the expected effect leads to inflated correlation

fMRI Processing Pipeline



Slide taken from SPM course

Realignment

Compensate for rigid transformations due to the patient's motion
→ motion parameters can be used as coregressors to account for changes in MR signal

Slice Timing Correction

Slices are captured sequentially. Thus, interpolate to a common timestep to get synchronized voxels

Normalization

Normalization to a standard space which is usually done indirectly by registering the image to some template

Smoothing

Apply gaussian smoothing to reduce noise, reduce number of comparisons and compensate residual misalignment

High-pass Filter

MR intensity can slowly drift between experiments. Use high-pass filter to remove this low frequency temporal component

General Linear Model

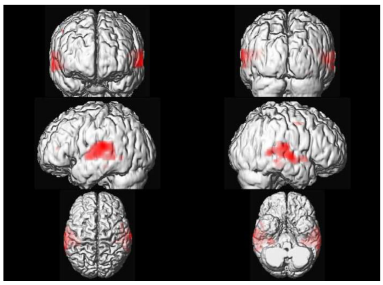
Inspect for each voxel given the measurements over a temporal interval if we had hemodynamic response.
→ fits BOLD time course to the signal and measures confidence
→ convolution with HRF smoothes out undersampled response

Statistical Inference

Perform voxel-wise statistical testing using RFT

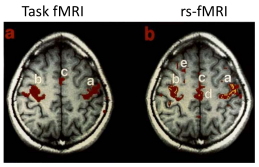
2D: Maximum intensity projection
→ visualize maximum of 3rd dimension

3D: Overlay activations with a 3D model of the cortical surface
→ after registration one can also take a generic model



Resting-State fMRI

Functional imaging over extended time periods with subjects told to think and do nothing
- → provides useful information about cognitive status
- → reduced experimental effort
- → same processing pipeline as fMRI
- → rs-fMRI shows similar patterns to fMRI



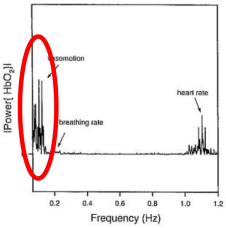
Motivation

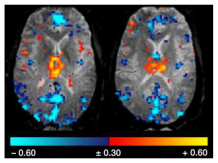
Localisationism: brain functions are localized to specific regions
- **Functional segregation:** Analysis of regionally specific effects
- **Functional integration:** Analysis of how diff. regions interact

Globalism: brain works as a whole and functions are split across it

Connectionism: brain is build up of simple connected units

BOLD signal: We use a bandpass filter to filter signals between 0.01-0.1 Hz
- → observed oscillating neural activity at this bandwidth



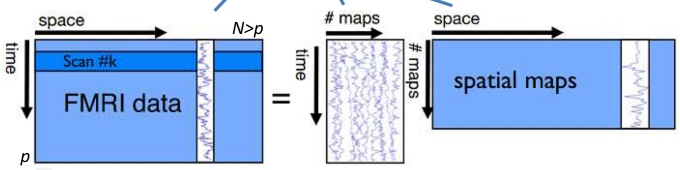


Independent Component Analysis

Decompos MR signal into spatially independent components that are activated according to the time course
- →exploratory analysis of unknown timecourse and spatial maps that created the signal

$$\mathbf{X} = \mathbf{AS} + \mathbf{E}$$

Gaussian additive noise



Assumptions:
1. Sources are statistically independent
2. Sources are zero-mean, non-gaussian distributed

Ambiguities:
1. Applying same permutations on A and S cancel out
2. Scaling columns of A and S by the inverse cancels out

Traditional ICA

Model: $\mathbf{X} = \mathbf{AS}$
- X: rows contain the recorded time signals
- S: rows contain the unknown source signals
- A: entry a_{ij} reflects influence of source j on timestep i

Goal: Estimate A to get an unmixing matrix that allows us to recover S

Solution:
1. Remove correlations, normalize variance in all dimensions
2. Whitening via spectral decomp: $\frac{1}{n}\mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^{\mathrm{T}} \Rightarrow \mathbf{U} = \mathbf{E}, \mathbf{\Sigma} = \mathbf{\Lambda}^{\frac{1}{2}}$
3. Perform SVD: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ or $\mathbf{W} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}$

→ Optimizing for V is left to make sources independent

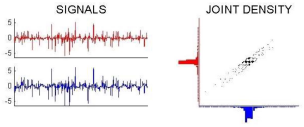
FastICA

Optimize one row at a time to maximize the non-gaussianity and solve for V

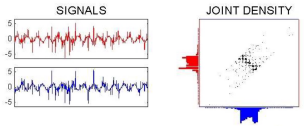
Maximize: $J(\mathbf{v}_i{}^T\tilde{\mathbf{x}}) = \sum_i k_i \left(\mathrm{E}[G_i(\mathbf{v}_i{}^T\tilde{\mathbf{x}})] - \mathrm{E}[G_i(\nu)]\right)^2$

- V: standard gaussian variable
- G_i: non-quadratic, not too fast growing function measuring gaussianity
- $\tilde{x}$: sphered data as before: $\tilde{\mathbf{x}} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{E}^{\mathrm{T}}\mathbf{x}$
- k_i: positive constants

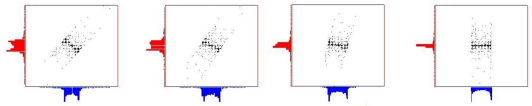
1. Initial signals and their joint and marginal densities



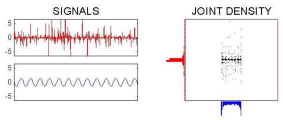
2. After initial whitening



3. Four iterations of optimizing **V** to achieve non-Gaussianity



4. Final result



Probabilistic ICA

Additionally model the error term that results when our system is over-determined (more timepoints than sources)

Model: $\mathbf{x}_i = \boldsymbol{\mu} + \mathbf{A}\mathbf{s}_i + \boldsymbol{\eta}_i$ for all voxels *i*

- x_i: timecourse of voxel i
- μ: spatial averages over all timepoints
- s_i: vector of independent, non-gaussian distributed random variables (spatial maps)
- η_i: gaussian noise vector

Number of sources q: has to be determined in advance
→ should equal the rank of the noise-free case
too small → unable to explain true variance in data
too large → networks break apart, meaningless maps

Solution: 1. Whitening: $\tilde{\mathbf{x}} = \left(\mathbf{\Lambda}_q - \hat{\sigma}^2\mathbf{I}\right)^{-\frac{1}{2}}\mathbf{E}_q^{\mathrm{T}}\mathbf{x}$

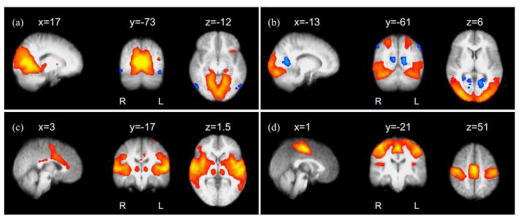
$\mathbf{\Lambda}_q$ and $\mathbf{E}_q$ contain first q eigenvalues and –vectors

Estimate $\hat{\sigma}^2 = \frac{1}{p-q}\sum_{l=q+1}^{p}\lambda_l$

2 Use FastICA to solve for V

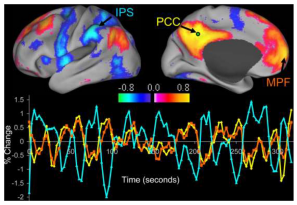
Resting State Network

Brain regions that consistently form independent components in rs-fMRI
→ can be affected by diseases

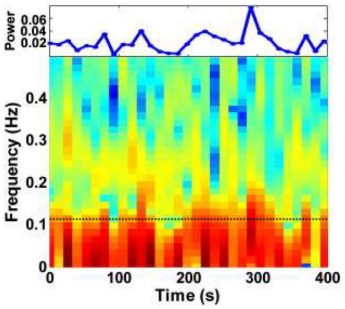


Default Mode Network

Regions / brain network that are more active at rest than at certain tasks



Cluster-based Analysis



Idea: Find clusters of voxels that have a similar timecourse in the frequency domain

Graph Cut: Partition graph into two clusters. The cut edge weights determine the cut's costs

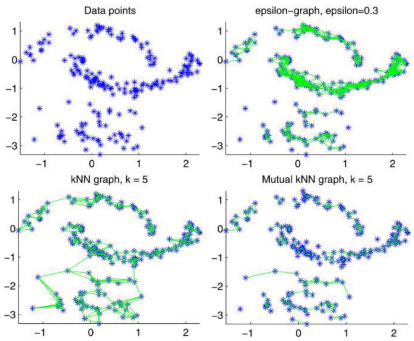
Ratio Cut: $\text{RatioCut} = \frac{\text{cut}(V_1, V_2)}{|V_1|} + \frac{\text{cut}(V_1, V_2)}{|V_2|}$

→ avoids splitting off small clusters

Normalized Cut: $\text{NCut} = \frac{\text{cut}(V_1, V_2)}{\text{vol}(V_1)} + \frac{\text{cut}(V_1, V_2)}{\text{vol}(V_2)}$

→ similarities between clusters are minimized

Neighborhood Graphs:



Graph for a given distance metric defining the neighborhood of each point

ε-Graph: connect points within ε

kNN: connect points to k NN

Mutual kNN: connect pair of points that are under each other's k NN

Spectral Clustering

Method to get the minimal NormCut by deriving the eigenvectors of the graph laplacian

→ exploit sparsity of laplacian during computation

→ relaxed computation of RatioCut & NormCut

Graph Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$ (unnormalized) and $\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$ (normalized)

W: affinity matrix
D: degree matrix

→ symmetric and positive semi-definite

Quadratic form: $f^T L f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$

f: indicator function, s.t.: $f_i = \sqrt{|\bar{A}|/|A|}$ if $v_i \in A$ and $f_i = -\sqrt{|A|/|\bar{A}|}$ if $v_i \in \bar{A}$

→ minimized for second smallest eigenvalue

Algorithm:

1. Compute Laplacian L = D-W
2. Compute the eigenvectors and take the eigenvectors corresponding to the k smallest eigenvalues

Binary: Threshold second smallest eigenvector

Multi:

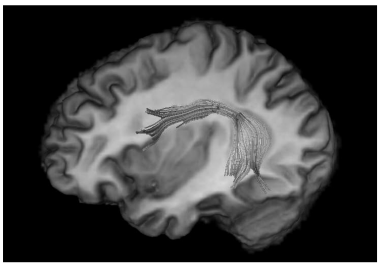
3. Stack eigenvectors in laplacian eigenmap
4. Perform standard clustering algorithm in that map

Spectral Gap: A stable clustering is characterized by k near-zero eigenvalues and a remarkable gap to the next larger eigenvalue

8. Diffusion MRI

Diffusion MRI

Investigate the microstructure of biological structure by measuring the diffusion of water in the brain. Water tends to be restricted in movement due to fibers.
→ traditional MRI does not provide sufficient resolution
→ reconstruction of spatial trajectories
→ reflects local fiber density and myelination



Molecular Diffusion

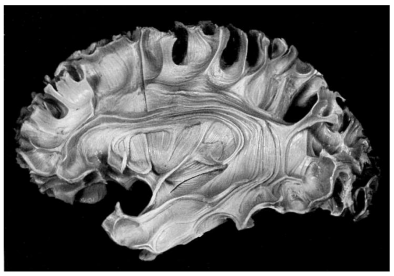
Due to thermal energy water molecule move in a random manner on a microscopic level.
This is restricted by cellular structure which can be measured.
→ happens without any concentration gradient (self-diffusion)

Diffusion coefficient: Measure of natural diffusion

White matter: $D \approx 1\text{-}5 \cdot 10^{-4}\ mm^2/s$

Klingler Dissection

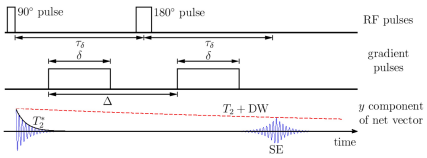
Traditional method to visualize white matter tracks by formalin-fixing and freezing the brain before scratching away tissue with a blund instrument

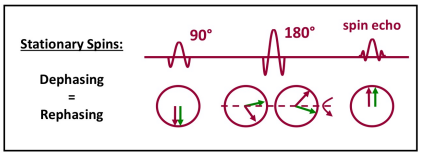


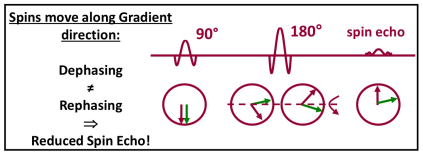
Stejskal-Tanner MR Sequence

MR sequence that makes diffusion along an applied field gradient visible

Effect: Stationary molecules experience the same phase diversion through the gradient which thus cancels out
Moving molecules experience diff. gradients such that the phase diversion does not cancel out.



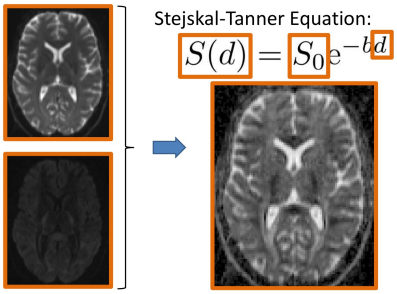




Signal Model: $\frac{S(g,\delta,\Delta)}{S_0} = \exp\left(-\gamma^2 g^2 \delta^2 \left(\Delta - \frac{\delta}{3}\right) D\right) \equiv \exp(-bD)$

Stejskal-Tanner eq.: $S(\mathbf{x}) = S_0 e^{-b\mathbf{x}^T \mathbf{D}\mathbf{x}}$

→ can be solved for di diffusivity the diffusivity



Diffusion Tensor Model

Models diffusivity through a quadratic function of the gradient:

$$S(D(\mathbf{x})) = S_0 e^{-bD(\mathbf{x})} \quad \text{with:} \quad D(\mathbf{x}) = \mathbf{x}^T \mathbf{D} \mathbf{x}$$

D: diffusion tensor, 3x3 symmetric matrix
b: sensitivity to changes corresp. to number of scan dir.

Eigenvalue decomposition: reveals main fiber directions
→ the eigenvector corresponding to the largest eigenvalue describes the main fiber direction

Estimation: Solve Stejskal-Tanner eq. for diffusion tensor and solve the resulting system of linear equations
→ usually more than 6 measurement and solution through least-squares

Linear system:

$$\sum_{k,l} \underbrace{\mathbf{x}_i^T \mathbf{D} \mathbf{x}_i}_{[\mathbf{D}]_{kl}[\mathbf{x}_i]_k[\mathbf{x}_i]_l} = -\frac{1}{b} \ln \frac{S(\mathbf{x}_i)}{S_0}$$

Mean Diffusivity: $(\lambda_1+\lambda_2+\lambda_3)/3$

→ average diffusion is typically much higher in ventricles

Fractional Anistropy:

$$\mathrm{FA} = \sqrt{\frac{3}{2}} \frac{\sqrt{(\lambda_1 - \mathrm{MD})^2 + (\lambda_2 - \mathrm{MD})^2 + (\lambda_3 - \mathrm{MD})^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

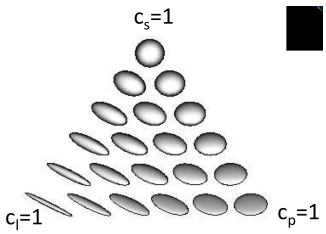
→ quantifies the degree of anistropy (non-uniform movement in spatial direction)
→ correlates with fiber density

Westin measure: Defines the extent to which a tensor ellipsoid is linear/ planar etc.

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$$

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$



Color coding: code spatial direction of diffusion

Tract-Based Spatial Statistics

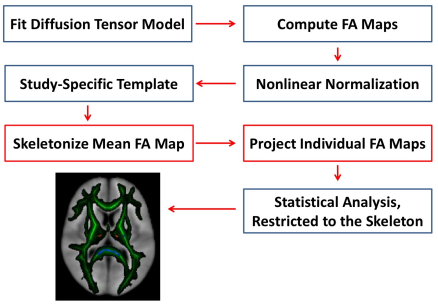
Motivation:

Natural approach: Normalization + Smoothing + Statistical Parametric Mapping

Problems:
1. FA varies strongly perpendicular to white matter tracts
   → non-interpretable results from imperfectly aligned brains
2. Avoid smoothing as it further decreases the resolution

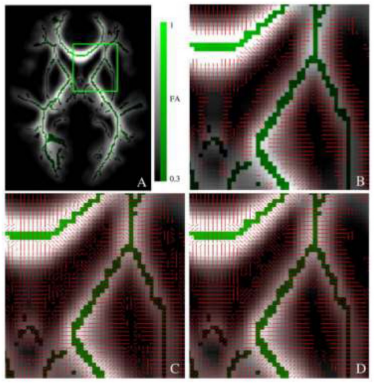
Idea: Instead of performing a voxel-based analysis, we extract the center (skeleton) of white matter structures described. We compute a medial surface through a group average over all FA maps and project the largest FA values to it.
→ often used in diffusion MRI

Pipeline:

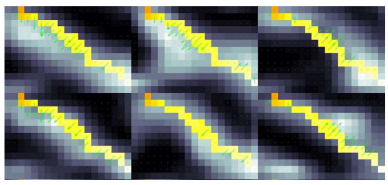


Skeletonization: Construct a quantized vector field perpendicular to the skeleton

1. Gradient direction g where g is sufficiently large
2. Direction e_3 of strongest concavity else
   → approximation of Hessian
3. Slight smoothing
4. Define skeleton where FA is locally maximal in direction of the vector field

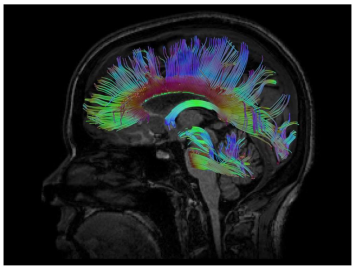


Projection: Find maximum for each pixel along vector field and project to it
→ ensure that we only have one-to-one projections



Problems:
- voxels from nearby bundles might be assigned to false brain parts
- projection might fail to compensate for missalignment
- results depend on choice of atlas

Fiber Tracking



Motivation:

Infer local fiber directions to reconstruct the trajectories of major white matter tracts

Deterministic tractography:

Proceed along a fixed tangent direction in each step.

Benefits:

- Successfully reconstructs many known bundles
  - Currently, only available method for in-vivo tract reconstruction
- Useful for neurosurgery and neuroscience
- Deterministic streamline-based methods are quite fast and simple

Limitations:

- Streamlines **do not** have a direct anatomical counterpart
  - Individual axons are much smaller
  - Fiber bundles are not point-to-point connections
- Issues of validation and false positives

Ingredients

Local fiber orientation estimation model

Model from before to estimate direction (Principal DTI eigenvector)

Numerical integration scheme

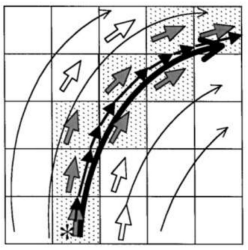
Model to evolve points along the fiber direction:

$$\dot{\mathbf{x}}(t) = \mathbf{v}(\mathbf{x}(t))$$

→ has to take into account that we are unable to estimate sign

Euler integration: $\mathbf{x}_{i+1} = \mathbf{x}_i + s\mathbf{v}(\mathbf{x}_i)$

Interpolation scheme



Integration scheme integrates outside the pixel space thus we need to interpolate to get fibers.

→ nearest neighbors easiest

→ interpolate diffusion tensor coefficients and re-compute fiber directions

Seeding scheme

Scheme to define starting points of fibers

→ seeds available for major white matter tracts

Track termination

Criterium to stop tracking of fibers

→ low anistropy, high curvature, leaving white matter mask

Track acceptance

Scheme to delete false tracks

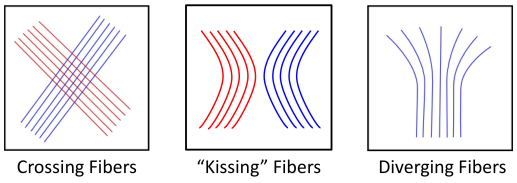
→ discard very short tracks

→ discard tracks ending in white matter

→ discard tracks not conform with anatomy

Multi-Fiber Tracking

Fibers tend to cross each other and fiber tracking has to account for it



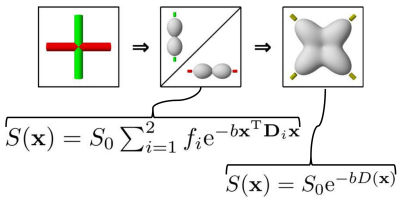
Multi-Fiber Tractography

Can be applied when we have multiple fiber estimates

1. Follow the most collinear with incoming direction
2. When in doubt branch
3. Regularize noisy multi-fiber estimates

Multi Tensor Models

Model $D(x)$ as a higher-degree polynomial
→ each fiber population contributes indep. to the signal
→ requires more gradient direction and higher b-values
→ impossible to recover isotropic parts of different fiber components



Ball and Stick Model

Model the diffusion tensor through a ball, model of center of gravity and overlap, and one or more sticks which model the independent fibers

Advantages:

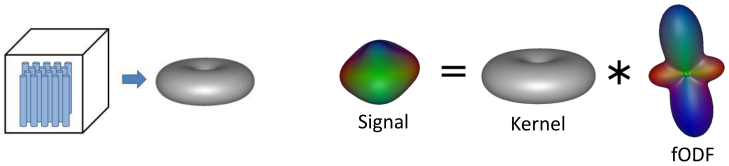
- natural generalization of the diffusion tensor model
- directly provides estimates of the fiber directions

Disadvantages:

- need to select the appropriate number of fibers
- model assumptions are restrictive
- nonlinear fitting is less reliable and efficient to linear methods

Spherical Deconvolution

Idea: If fibers contribute the same MR signal up to an orientation the overall signal can be represented through a convolution of a orientation distribution function (fODF) on a kernel that represents the signal from a single fiber.



Spherical convolution:

Intuition of spherical convolution between the fiber ODF $F(\theta, \phi)$ and an axially symmetric kernel $R(\theta)$:

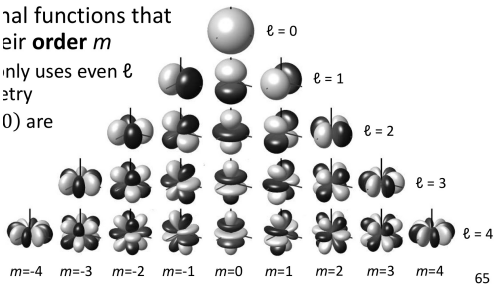
- Rotate a copy of R to align its axis with any direction (θ', ϕ') on the sphere
- Scale its contribution by the local value of $F(\theta', \phi')$
- Integrate the contributions of all rotated and scaled copies

The resulting **convolution integral** can be written as

$$S(\theta, \phi) = \int_0^{2\pi} \int_0^{\pi} F(\theta', \phi')\, R(\gamma') \sin(\theta')\, d\theta'\, d\phi'$$

Spherical Harmonics

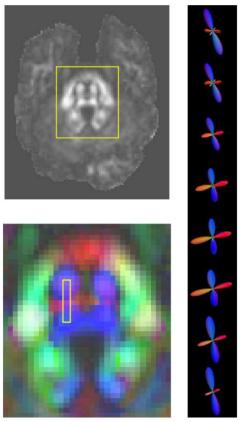
Provides a set of orthonormal basis functions on the sphere that is analogous to the Fourier basis



65

Spherical Convolution Algorithm

1. Linear least-squares fit of spherical harmonics to the HARDI measurements in each voxel
2. Estimate single-fiber response from putative single-fiber voxels (high FA in DTI analysis)
3. Compute fODFs via SH convolution theorem
   - *Main challenge:* Division by small values for large ℓ greatly amplifies measurement noise. Solved by low-pass filtering the result
4. fODF maxima are taken as fiber directions, their magnitude as volume fractions



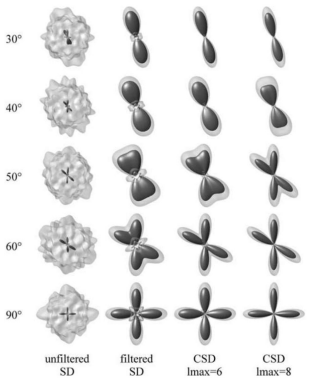
Constrained Spherical Convolution

Non-negativity constraint permits omitting the low-pass filters in fODF
→ can increase angular resolution

In each voxel:

1. Perform filtered deconvolution to obtain an initial fODF $\mathbf{f}_0$
2. Create a set N_0 of directions with negative fODF values based on checking 300 uniformly distributed directions
3. Iterate until convergence:
   a. Solve the following regularized least squares problem:
   $$\mathbf{f}_{i+1} = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{f} - \mathbf{b}\|^2 + \lambda^2 \|\mathbf{L}_i \mathbf{f}\|^2$$
   - $\mathbf{A}$ convolves fODF $\mathbf{f}$ with the single-fiber kernel and evaluates the result in all measurement directions, $\mathbf{b}$ contains the diffusion-weighted measurements
   - $\mathbf{L}_i$ evaluates $\mathbf{f}$ in all directions contained in N_i, λ is a regularization weight

   b. Update N_{i+1} based on $\mathbf{f}_{i+1}$



Advantages:

- does not require pre-defined number of fiber compartments
- sharpness of fODF peaks models spread in addition to crossings

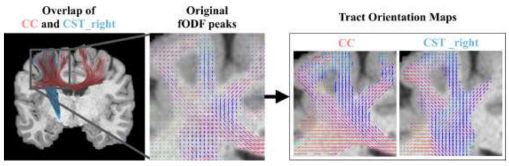
Disadvantages:

- strong assumption that fiber response is uniform
- deconvolution is numerically ill-posed

Machine Learning Based

Tract orientation maps:

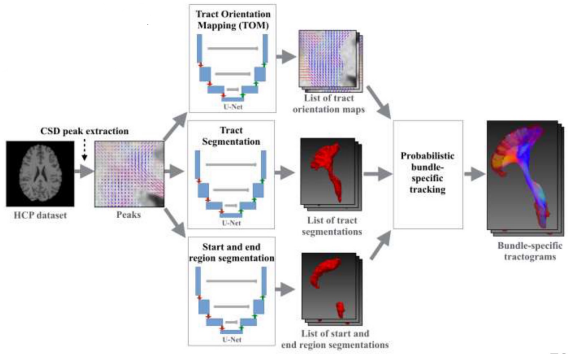
Bundle-specific maps that give a tract direction for each voxel
→ tends to make bundles too wide



TractSeg

Additionally predict a mask for a tract to counteract wideness of TOMs

TOM-based pipeline:



9. Machine Learning Based Neuroimaging

Supervised Learning

Goal: Learn a function $f(x)=y$ mapping a feature vector x to a label y via training data $(x, f(x))$

Classification: y is discrete

Regression: y is continuous

Learning: Minimize the risk / empirical risk over the training data via a loss function

Training data: used for optimizing the model parameters

Validation data: used to tune hyperparameters

Test data: used to evaluate the method

Cross-Validation

Split data in n-folds, test on one and train on rest. Perform n runs to estimate risk.
→ used when data is limited

Training	Training	Training	Training	Test

Evaluation

		Prediction $f(\mathbf{x})$	
		Positive	Negative
True Label y	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Accuracy: $ACC = \frac{TP+TN}{TP+TN+FP+FN}$

Precision: $P = \frac{TP}{TP+FP}$

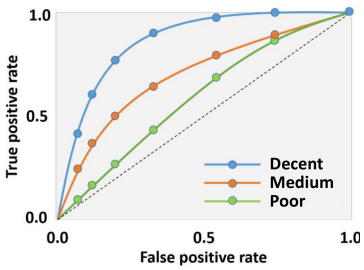
Recall: $R = \frac{TP}{TP+FN}$

F-score: $F = 2\frac{P \cdot R}{P+R}$

ROC-curve: reflects the effect of varying thresholds

Recall = **True-Positive-Rate** $TPR = R = \frac{TP}{TP+FN}$

False-Positive-Rate $FPR = \frac{FP}{TN+FP}$



AUC: area under ROC-curve

Regression Models

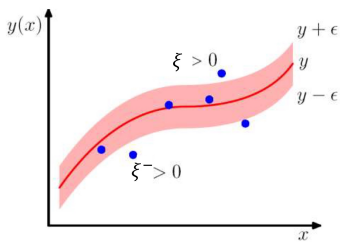
Linear Regression: $\|\mathbf{Ax}-\mathbf{b}\|^2$

Ridge Regression: $\|\mathbf{Ax}-\mathbf{b}\|^2 + \alpha\|\mathbf{x}\|^2$

Kernel Ridge Regression: same as ridge with kernel

Definition: Σ-insensitive loss
Define Σ-tube around function where points do not contribute to the error:

$L=\max(0,|y-y'|-\varepsilon)$

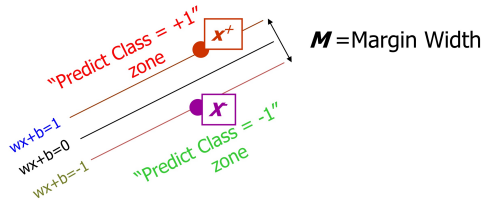


Slack Representation:

$$y_n \le f(x_n) + \varepsilon + \xi_n$$
$$y_n \ge f(x_n) - \varepsilon - \xi_n^-$$

Support Vector Machines

Idea: Finding a linear classifier is ill-posed. To find the potentially best generalizing linear classifier, we not only want to separate the data but also maximize the margin



Classification Function: $f(\mathbf{x},\mathbf{w},b) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$

Goals:

1) **Correctly classify all training data**

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1 \quad \text{if } y_i = +1$$
$$\mathbf{w}^T\mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1$$
$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \text{for all i}$$

2) **Maximize the Margin** $M = \frac{2}{\|\mathbf{w}\|}$

same as minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

Support Vectors: Vectors on the margin supporting it

Primal Form: **Minimize** $\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$

subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$

Lagrangian: $L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N}\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1)$

Dual Problem: **Maximize** $L(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ s.t. $\Sigma\alpha_i y_i = 0$

→ set derivative of Lagrangian to zero and solve for w to eliminate in formulation

Soft-Margin: Find **w** and *b* such that

$\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized and for all $\{(\mathbf{x}_i, y_i)\}$

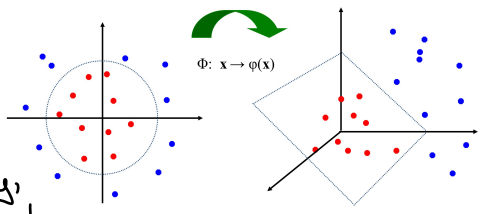
$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all *i*

→ allow misclassifications

Large C: behaves like hard-margin SVM

Small C: Tolerate more misclassifications

Kernel Trick: Map the data to a higher dimensional feature space that promises linear separability. Instead of computing the mapping, we only compute the dot-product in the feature space



Kernel: $K(\mathbf{x}_i,\mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ $\quad \Phi: \mathbf{x} \rightarrow \phi(\mathbf{x})$

Linear: $K(\mathbf{x}_i,\mathbf{x}_i) = \mathbf{x}_i^T\mathbf{x}_i$ **Polynomial:** $K(\mathbf{x}_i,\mathbf{x}_j) = (1 + \mathbf{x}_i^T\mathbf{x}_j)^p$

RBF: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2}$

Regression:

Primal Form: Minimize $\frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^-)$

such that $y_n \leq f(x_n) + \varepsilon + \xi_n$ and $\xi_n \geq 0$

$y_n \geq f(x_n) - \varepsilon - \xi_n^- \quad \xi_n^- \geq 0$

Dual Form: Maximize:

$$Q(a,a^-) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - a_n^-)(a_m - a_m^-)k(x_n,x_m) - \varepsilon\sum_{n=1}^{N}(a_n + a_n^-) + \sum_{n=1}^{N}(a_n - a_n^-)y_n$$

$$0 \leq a_n, a_n^- \leq C$$

Prediction: $f(x) = \sum_{n=1}^{N}(a_n - a_n^-)k(x,x_n) + b$

Feature Selection Strategies

Embedded Methods: Integrate feature selection into the training process

Wrapper Methods: Systematic selection through training and evaluation of feature subsets

Sequential Backward Selection:
Starting with all features, iteratively remove features to achieve best cross-validation score

Recursive Feature Elimination:
Remove features based on importance score to improve computational efficiency
→ absolute value of corresponding w in SVM

Filter Methods: Select features agnostic to ML method by assessing its amount of information
→ e.g. compute score indicating its usefullness
→ typically ignores feature redundancy and complementarity

Linear Correlation: $|\rho| = \left|\frac{\mathrm{cov}(x_i, y)}{\sigma_{x_i}\,\sigma_y}\right|$
→ regression

t/F Test Statistics: Comparison of group of features
→ classification

Mutual Information: $I(X,Y) = H(X) + H(Y) - H(X,Y)$

Feature Normalization

Normalize different features to bring to same magnitude to avoid numerical issues and dominance of single features.

Standardize: zero-mean with unit variance

Min-Max Scaling: Linearly scale to $[0,1]$

Variational Autoencoder

Combination of encoder and decoder to first suppress the input to a latent representation and then reconstruct the image from that representation.

Latent Diffusion Model

Suppress input to latent representation, encode noise image with it and iteratively remove noise to learn diffusion process