Lecture 13
Segmentation and Scene Understanding

Chris Choy, Ph.D. candidate
Stanford Vision and Learning Lab (SVL)
http://chrischoy.org

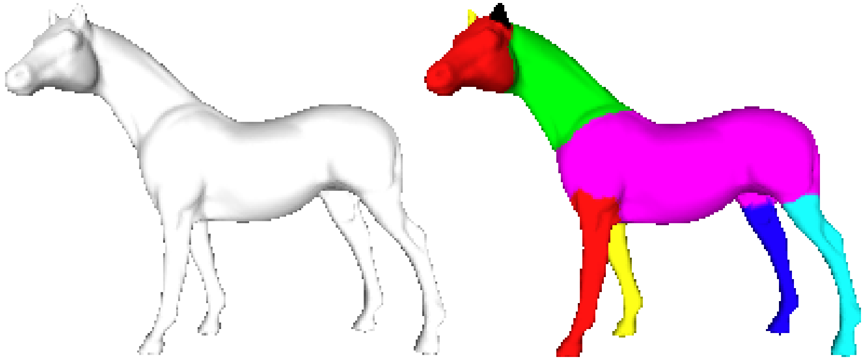
Understanding a Scene

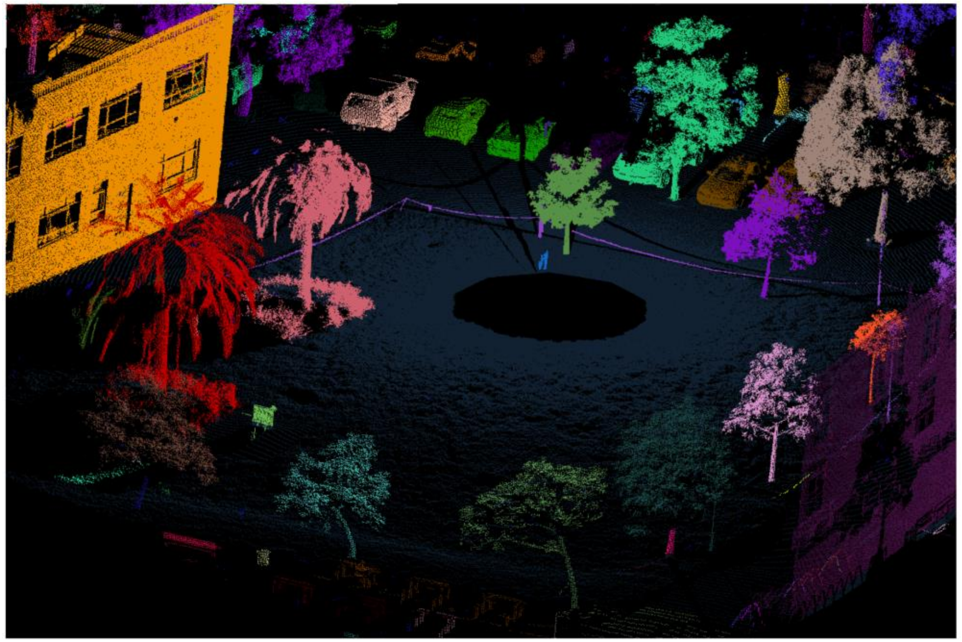
- Objects
  - Chairs, Cups, Tables, etc.…
  - Bounding boxes and labels
- Amorphous objects
  - Sky, Lawn, Background, etc.…
  - ???

Image Segmentation

- One way to represent an image using a set of **components**
- **Components** share common properties
- Properties can be defined at different levels of abstraction

Segmentations 1D and 3D



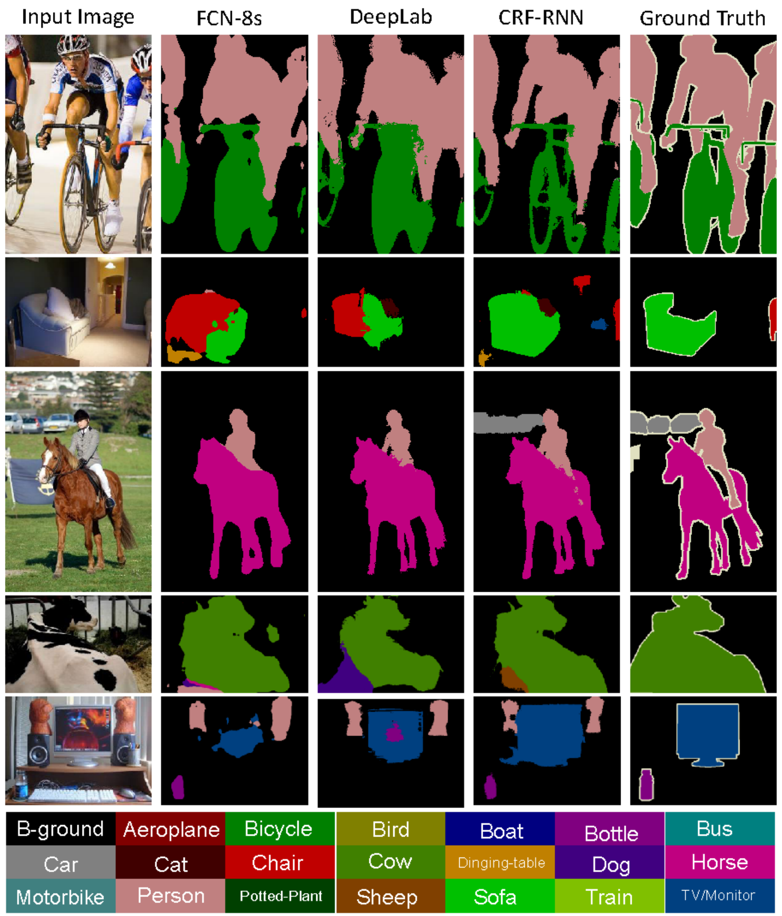


- Sentence segmentation, topic segmentation

Table of Contents

- Types of Segmentations
- Clustering-based Segmentation
  - K-means
  - Mean Shift
- Graph-based Segmentation
  - Normalized Cut, Spectral Clustering
  - Conditional Random Field
- Supervised Segmentation
  - Feature learning
  - Fully Convolutional Neural Network (FCNN)
  - Probabilistic Graphical Model (CRF) + FCNN
  - Spectral Clustering + FCNN
- Example code

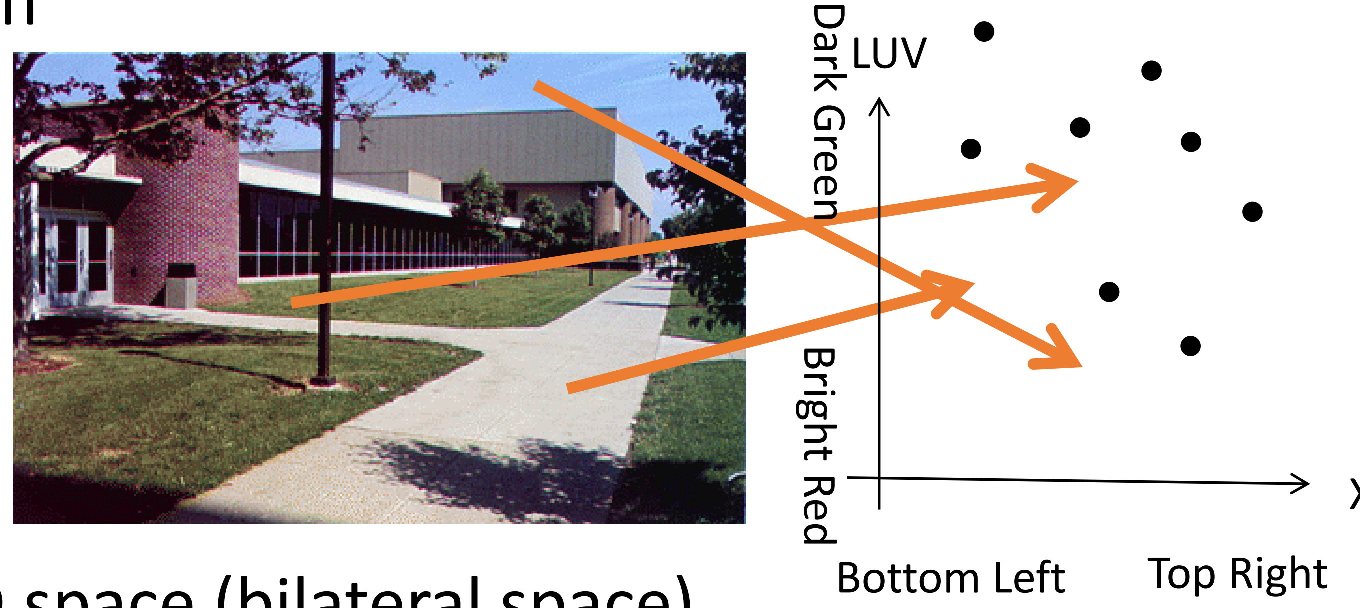
Segmentation using Neural Networks



- Why should we learn old techniques that are beaten by neural networks?
- Neural Networkism
  - A social phenomenon or belief that giant neural networks can solve everything
- Ex. Consistency in a neural network

Clustering-based Segmentation

- Clustering-based segmentation
  - K-means clustering
  - Non-parametric Bayesian
  - Energy-based methods …



- Each pixel = data point in a 5D space (bilateral space)
  - XYRGB or XYLUV
- Should we only use the 5D feature? (Hint: kernel)

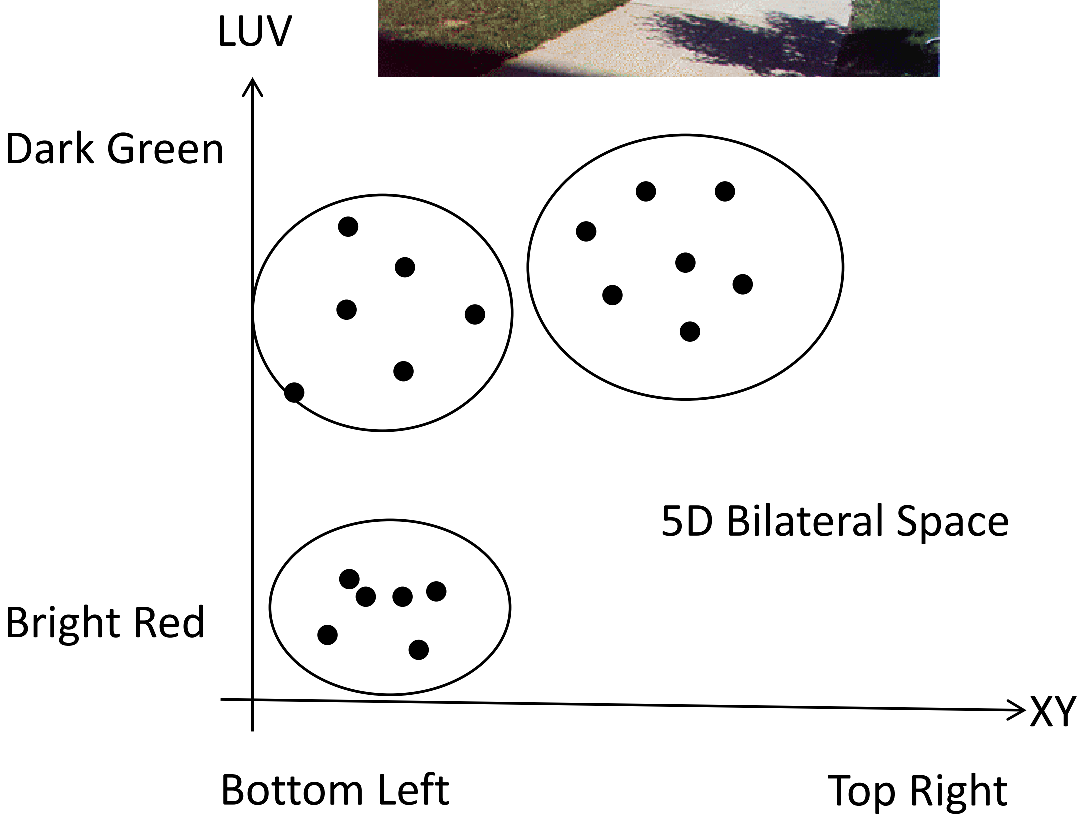
K-means Clustering

$$E(\mathbf{X}, \mathbf{S}) = \sum_{i}^{K} \sum_{x \in S_i} ||x - \mu_i||$$

- Minimize the sum of distance to the centroid for all clusters
- NP-hard (Dasgupta et al. The hardness of k-means clustering)
- Heuristic algorithm
  - Random initialization
  - Repeat:
    - Assignment: find the cluster ID for all point
    - Update centroids

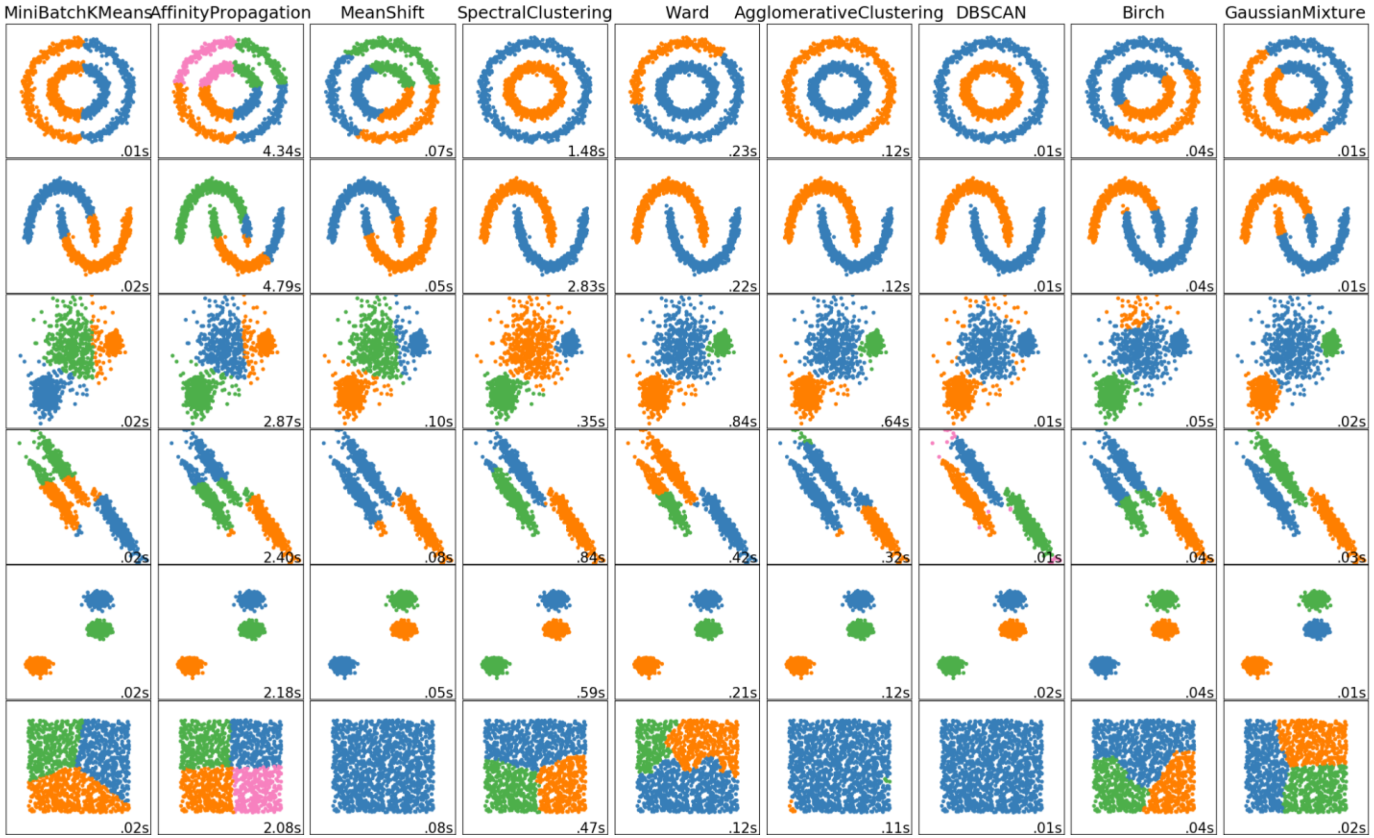
Clustering-based Segmentation

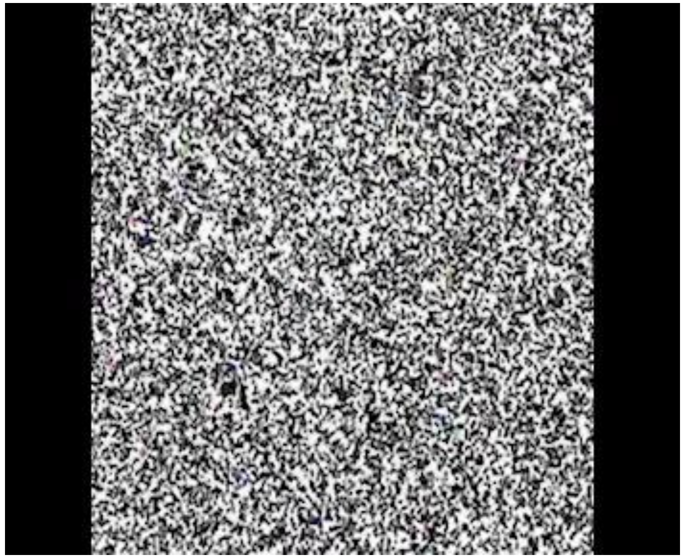
- K-means:
  - Soft assignment: **Multi-modal Gaussian** fitting
- Non-parametric Clustering:
  - Affinity Propagation
  - DBSCAN
  - Mean Shift
  - …

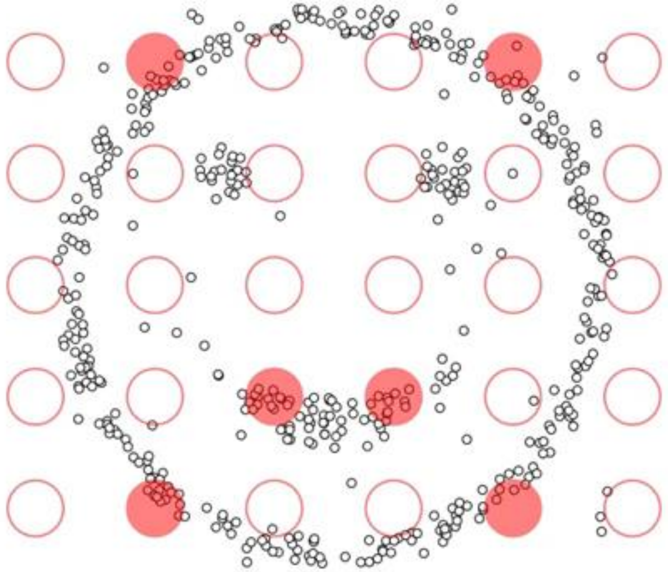


(a) K=50 (b) K=100 (c) K=200 (d) K=300 (e) K=400

Clustering Methods







sklearn.cluster

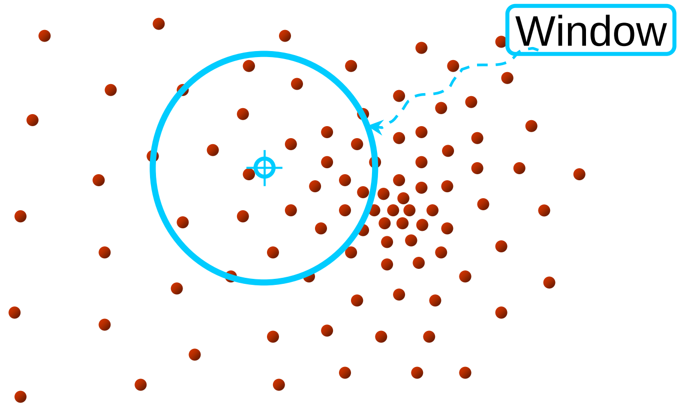
Mean Shift

- Non-parametric, iterative clustering method

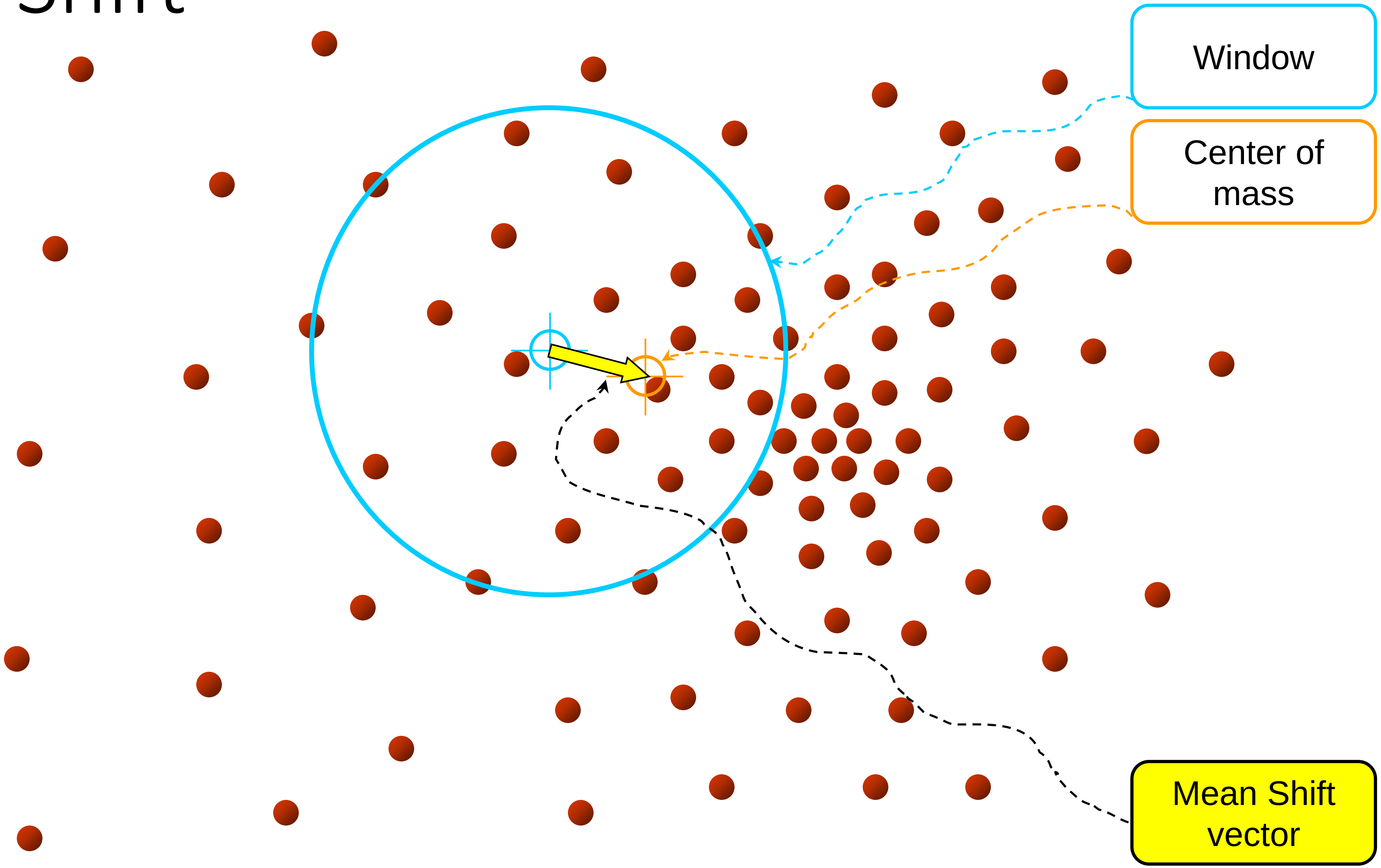
  Structure of the model is not fixed

  Improves an initial guess by sequentially updating it

- Seeks modes or local maximum (plural of maxima) within a window
- Algorithm:
  - Starts from over sampled initial centroids
  - Repeat until convergence
    - Iteratively update centroids
    - Remove overlapping centroids if too close

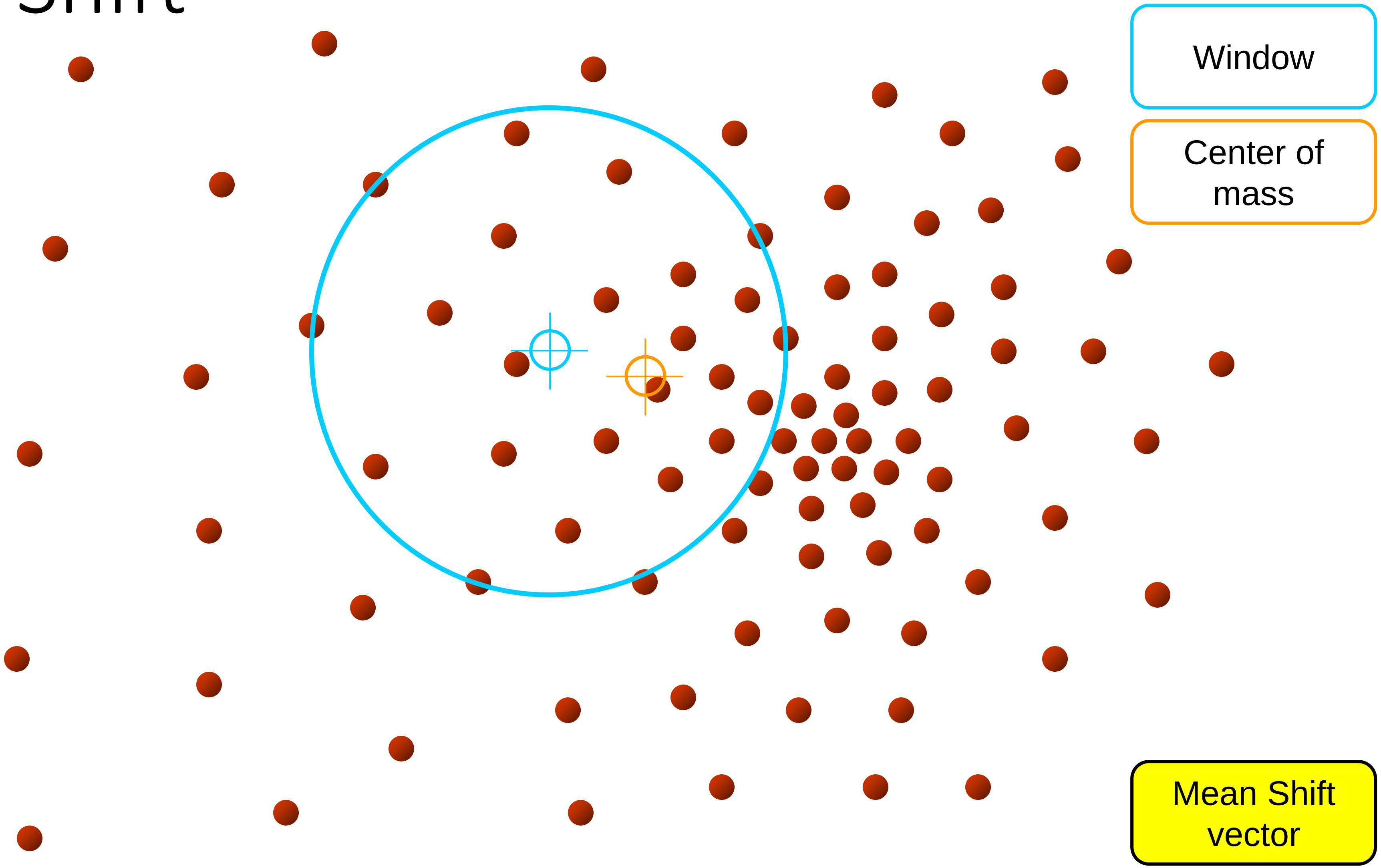


Mean Shift



Slide by Y. Ukrainitz & B. Sarel

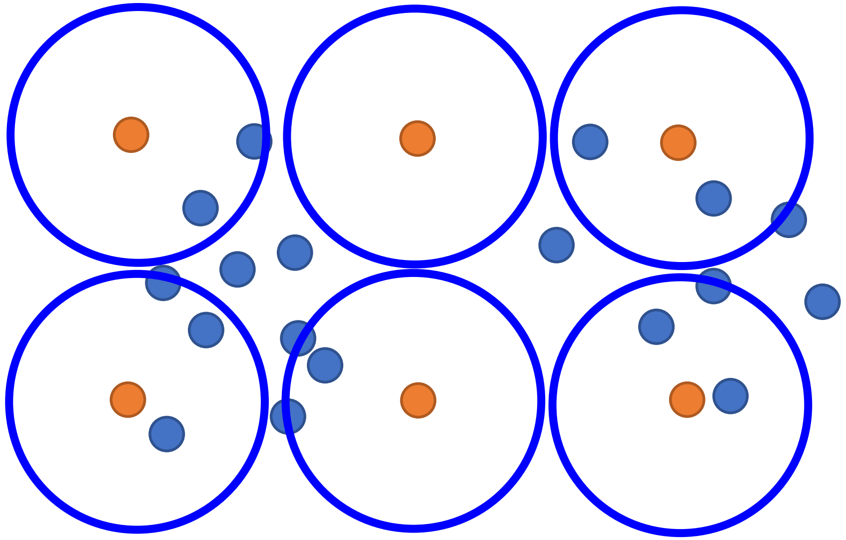
Mean Shift



Slide by Y. Ukrainitz & B. Sarel

Mean Shift

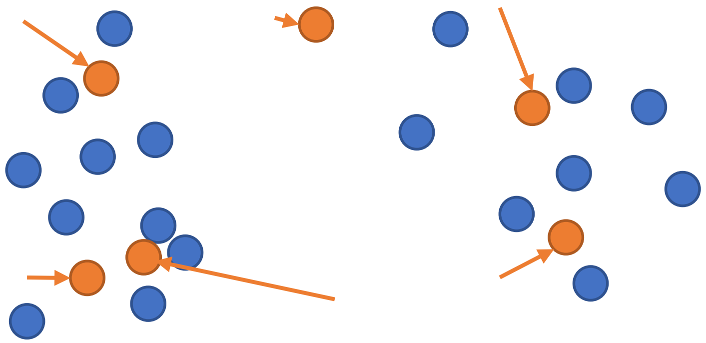
1. Initialize centroids (tessellation of space with windows)



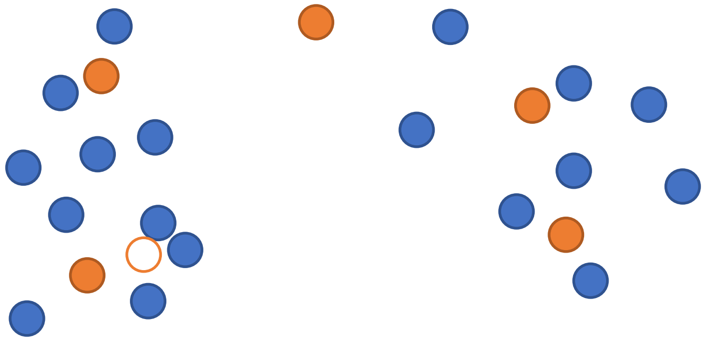
2. Update centroids

$$\mathbf{c} = \sum_i w(\mathbf{x}_i, \mathbf{c})\mathbf{x}_i$$

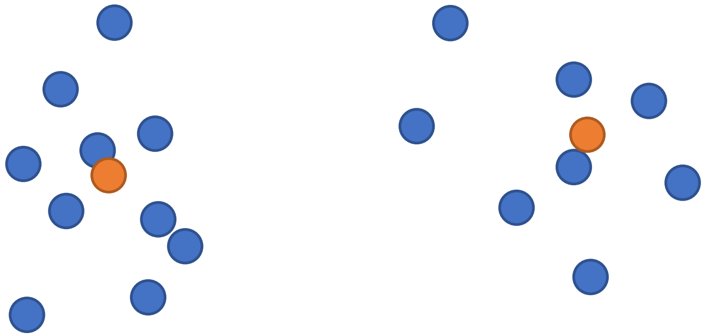
$$\sum_i w(\mathbf{x}_i, \mathbf{c}) = 1$$



3. Merge centroids

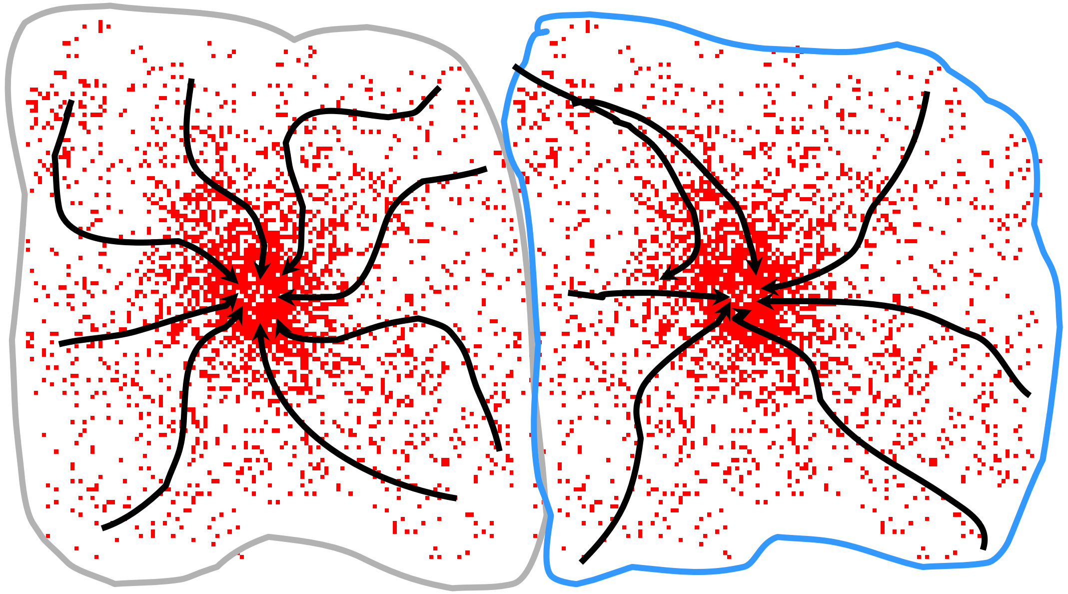


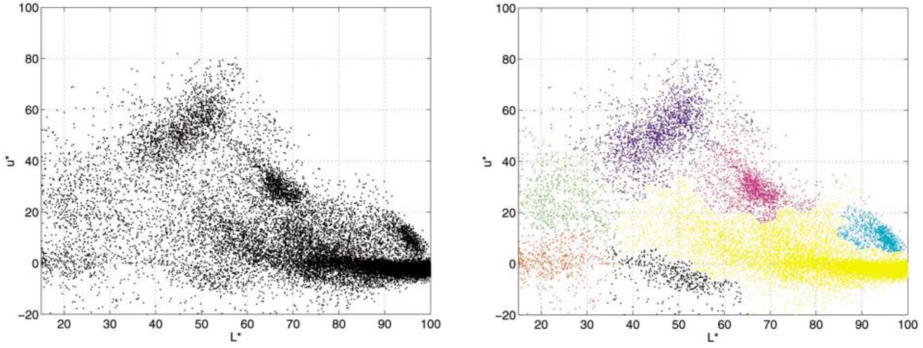
4. Repeat 2, 3 until convergence



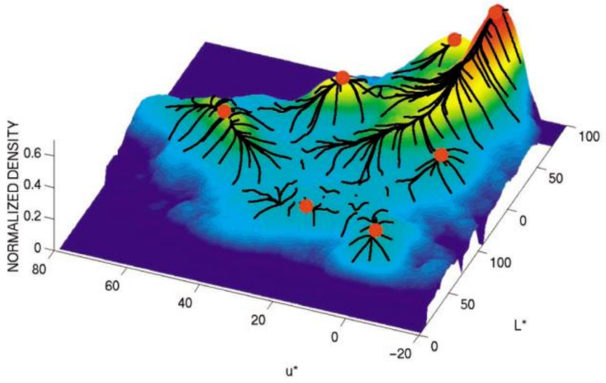
Mean Shift: Attraction Basin

- Attraction basin: the region for which all trajectories lead to the same mode
- Cluster: all data points in the attraction basin of a mode





(a) (b)



Slide by Y. Ukrainitz & B. Sarel

Demo

- git clone http://github.com/chrischoy/segmentation_lecture
- cd segmentation_lecture
- (sudo) pip install –r requirements.txt
- python kmeans.py
- python meanshift.py

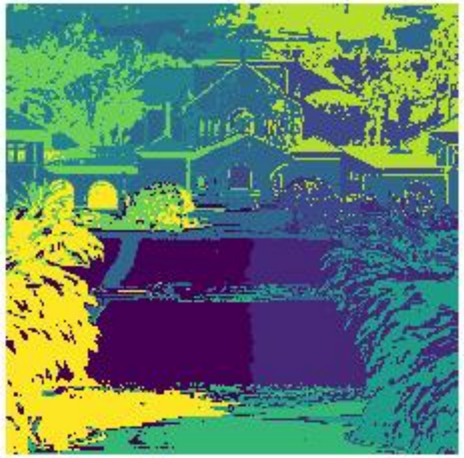
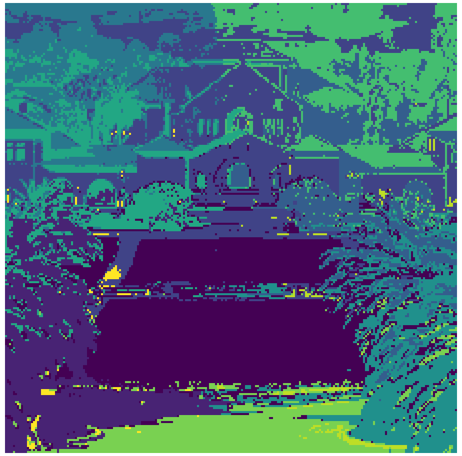




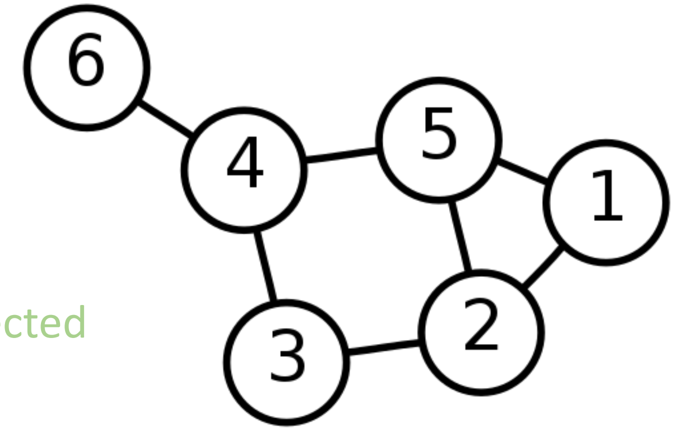
Table of Contents

Graph-based Segmentation

- Min-cut
- Normalized Cut
- Spectral Clustering
- Probabilistic Graphical Model
  - Conditional Random Field (CRF)
  - Markov Random Field (MRF)
  - Potts-model

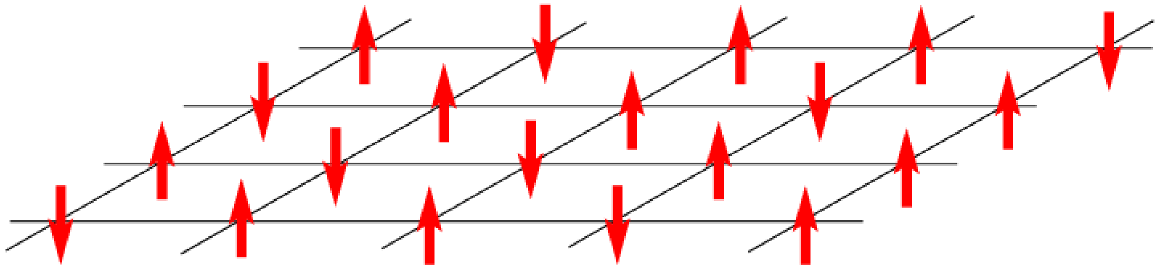
Graphs on Image

- Graph: Data structure consisting of nodes, edges
- Weighted Undirected Connected Graph



Edges have associated weights   Edges are bidirectional   Every pair of nodes is connected

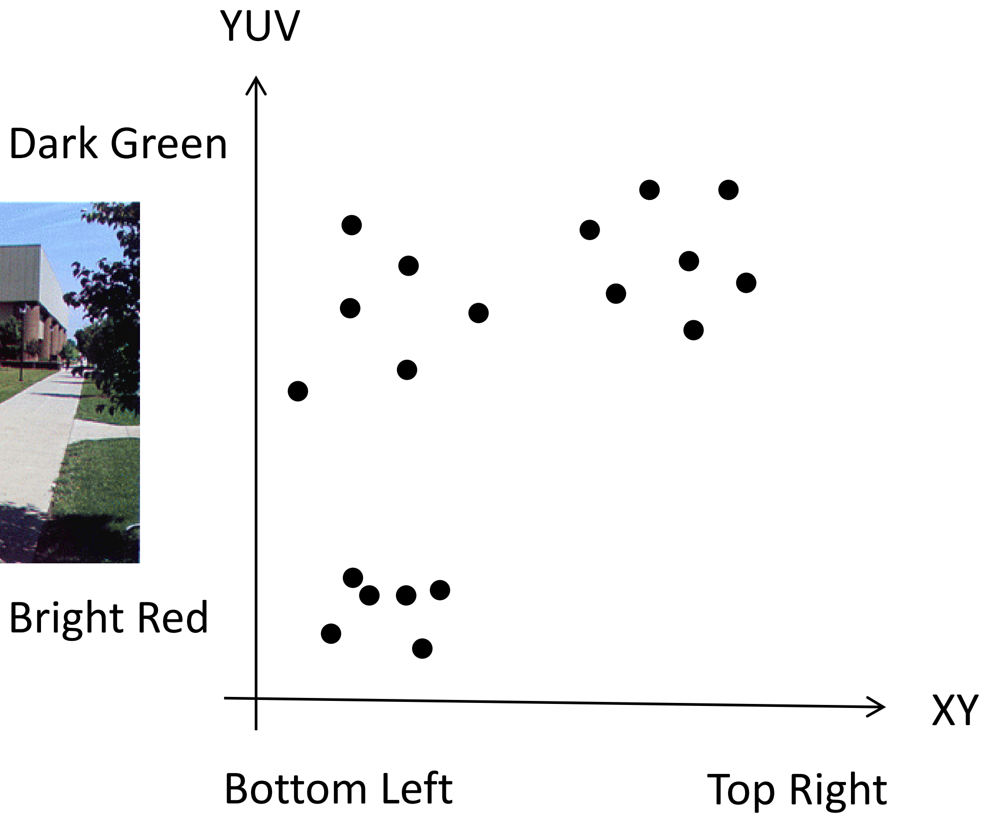
- Ising Model

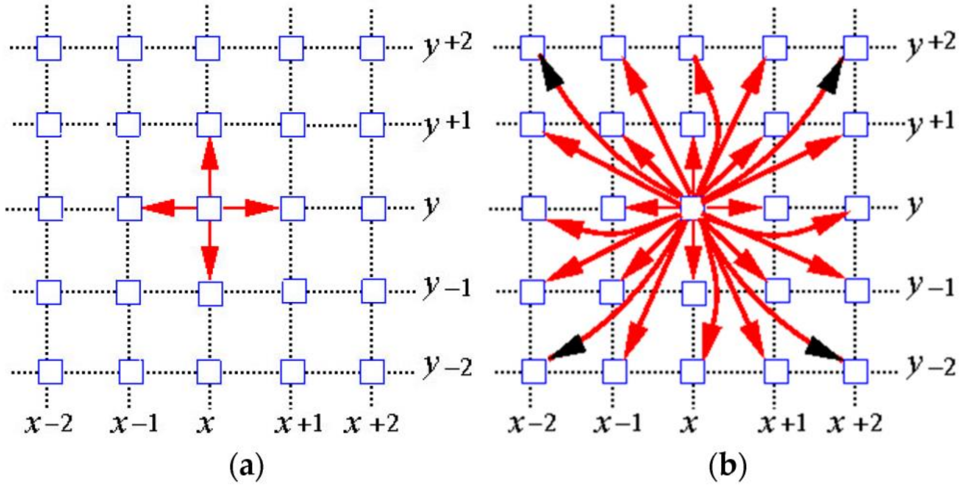


2–D Ising Model

- Probabilistic Graphical Models
  - Markov Random Field and Conditional Random Field

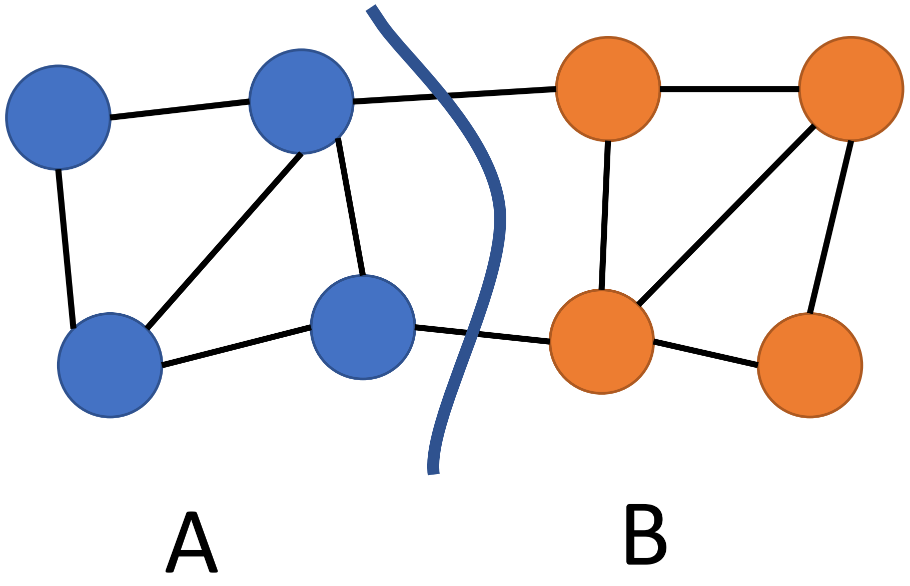
More Graphs on Image





Graph-cut, Min-cut

- Represent features and their relationships using a weighted graph
  - Node ← every pixel, superpixel
  - Edge ← Affinity or similarity between two nodes
    - Affinity can be innerproduct between features (color) or RBF kernels
- Cut the graph to subgraphs



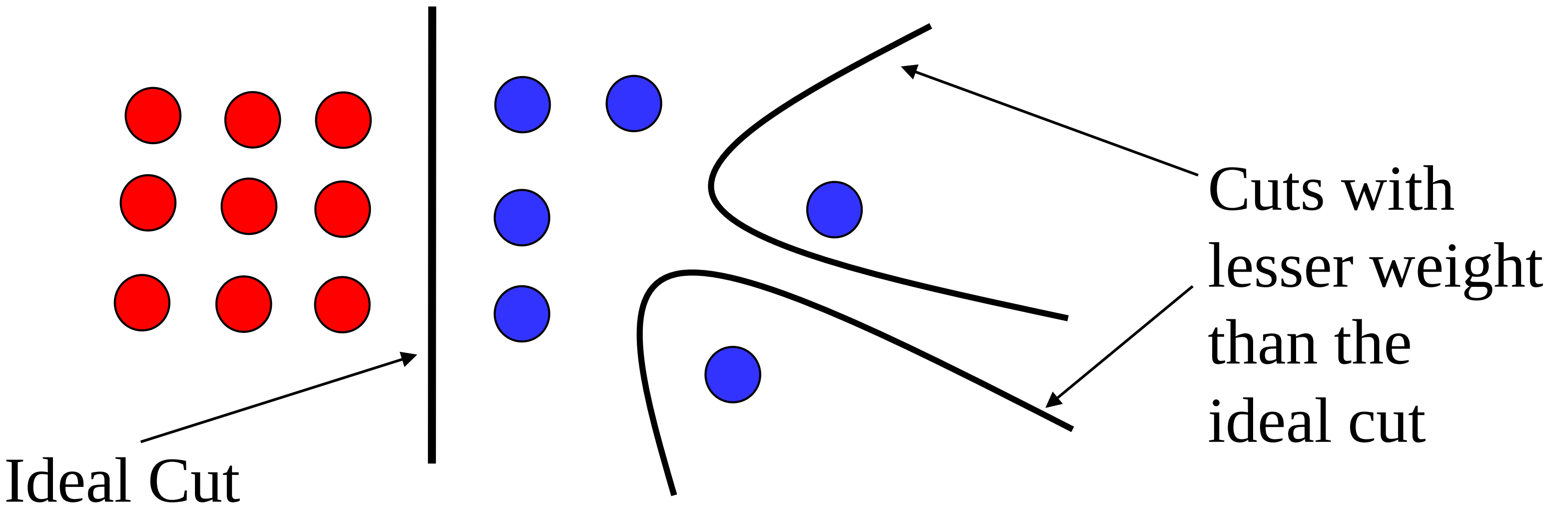
$$G = \{V, E, W\}$$

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} W_{ij}$$

Normalized Cut

- Min-cut will favor isolated nodes

$$G = \{V, E, W\}$$



- Normalize cut by its all weights

$$\mathrm{cut}(A, B) = \sum_{i \in A, j \in B} W_{ij} \qquad \mathrm{Ncut}(A, B) = \frac{\mathrm{cut}(A, B)}{\sum_{i \in A, j \in V} W_{ij}} + \frac{\mathrm{cut}(A, B)}{\sum_{i \in B, j \in V} W_{ij}}$$

R-ary Normalized Cut

- Disjoint sets $A = (A_r)_{r\in\{1,\ldots,R\}}$ $\quad \cup_r A_r = V$

$$\mathrm{Ncut}(A,B) = \frac{\mathrm{cut}(A,B)}{\sum_{i\in A, j\in V} W_{ij}} + \frac{\mathrm{cut}(A,B)}{\sum_{i\in B, j\in V} W_{ij}}$$

$$\mathrm{cut}(A,B) = \sum_{i\in A, j\in B} W_{ij}$$

$$C(A,W) = \sum_{r=1}^{R} \frac{\sum_{i\in A_r, j\in V\setminus A_r} W_{ij}}{\sum_{i\in A_r, j\in V} W_{ij}}$$

$$e_r \in \{0,1\}^{|V|} \qquad D = \mathrm{diag}(W1)$$

$$c(e,W) = \sum_{r=1}^{R} \frac{e_r^T (D-W) e_r}{e_r^T D e_r}$$

Bach & Jordan, NIPS'03

Normalized Cut and Spectral Clustering

$$c(e, W) = \sum_{r=1}^{R} \frac{e_r^T (D - W) e_r}{e_r^T D e_r}$$

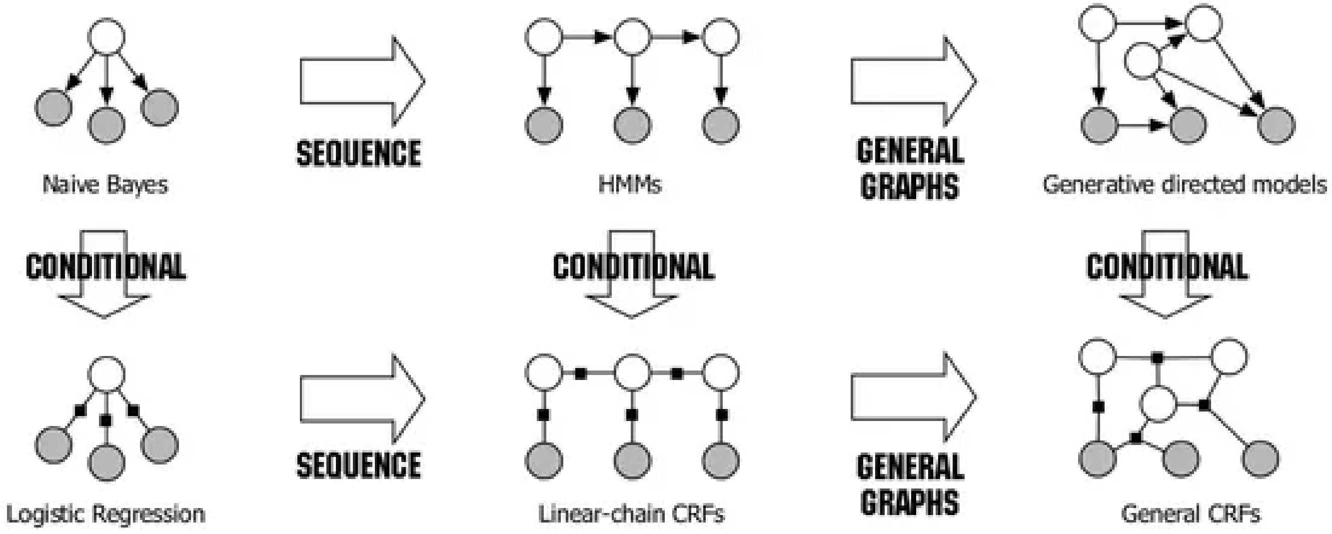
- Finding the optimal $e_r \in \{0, 1\}^{|V|}$ is NP-hard
- Continuous relaxation
  - Relax discrete variables to have intermediate values

$$c(Y, W) = R - \mathrm{tr} Y^T D^{-1/2} W D^{-1/2} Y \qquad Y = [e_1, e_2, ..., e_R] \qquad Y^T Y = I$$

- The solution is the sum of the R largest eigenvalues of $D^{-1/2} W D^{-1/2}$
- From the second smallest eigenvector $I - D^{-1/2} W D^{-1/2}$
  - Normalized Laplacian
- R-ary clustering
  - Hierarchically discretize corresponding eigenvectors

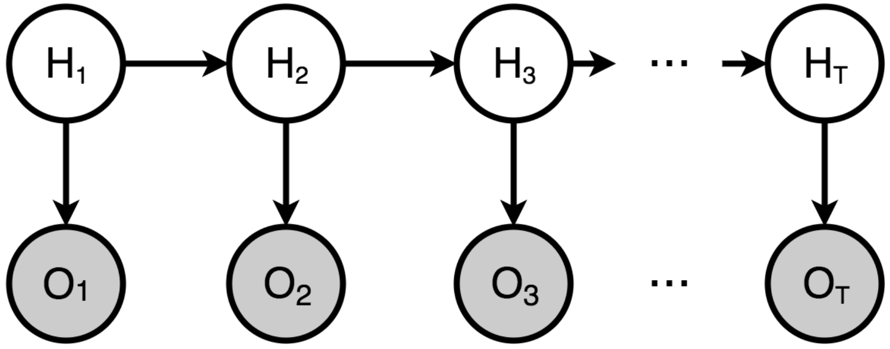
Probabilistic Graphical Model (PGM)

- Graphical representation of conditional dependence structure
- Markov Random Field
- Conditional Random Field



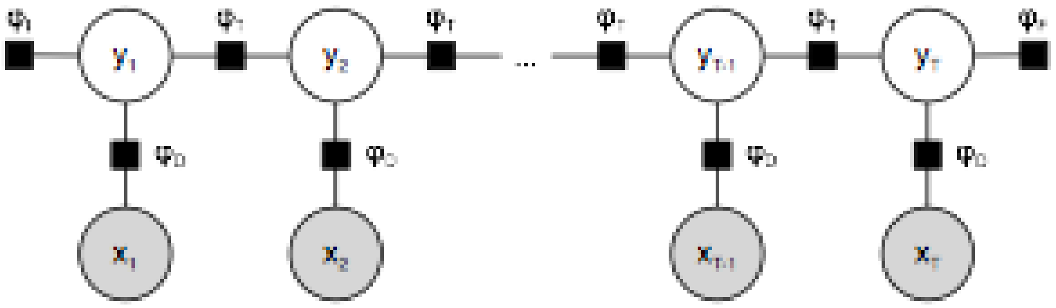
1D PGMs

- Hidden Markov Model



$$p(\mathbf{H}, \mathbf{O}) = p(H_1) \prod_t^{T-1} p(H_{t+1}|H_t) \prod_t^{T} p(O_t|H_t)$$

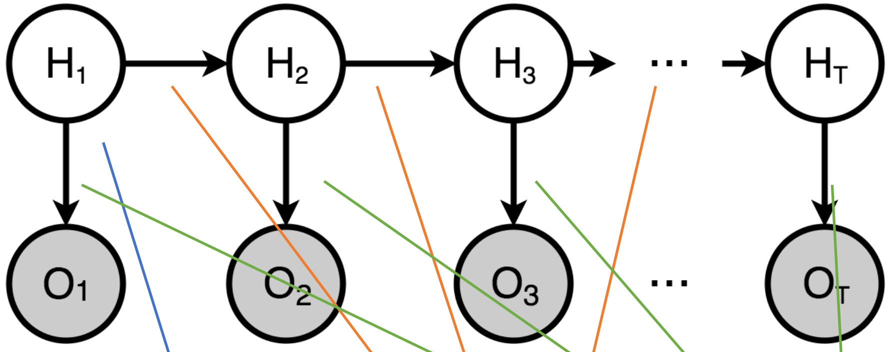
- Conditional Random Field



$$E(\mathbf{X}, \mathbf{Y}) = \sum_t^T \phi(X_t, Y_t) + \sum_t^T \psi(Y_t, Y_{t+1})$$

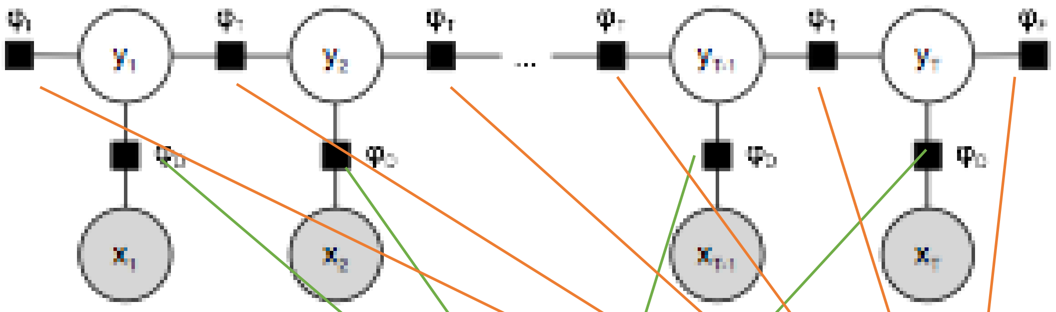
1D PGMs

- Hidden Markov Model



$$p(\mathbf{H},\mathbf{O}) = p(H_1)\prod_t^{T-1} p(H_{t+1}|H_t)\prod_t^{T} p(O_t|H_t)$$

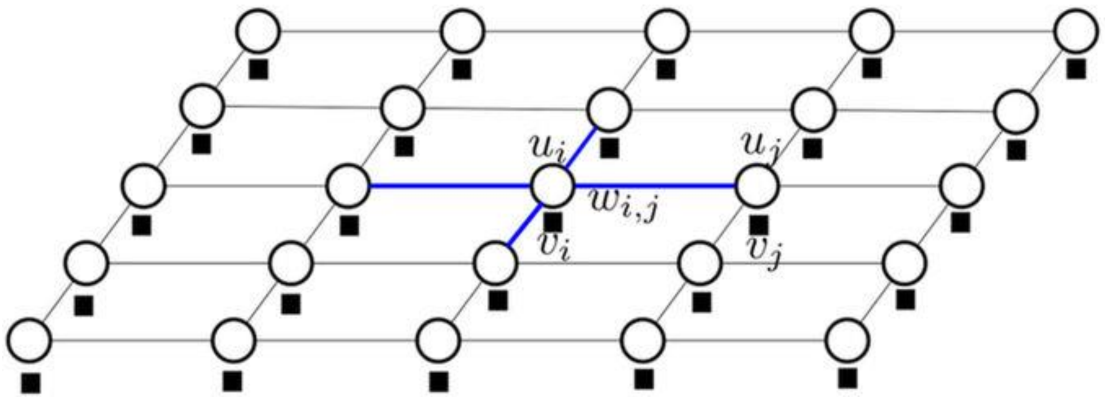
- Conditional Random Field



$$E(\mathbf{X},\mathbf{Y}) = \sum_t^T \phi(X_t, Y_t) + \sum_t^T \psi(Y_t, Y_{t+1})$$

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z}\exp(E(\mathbf{X},\mathbf{Y}))$$

2D Conditional Random Field



- Unary potential
  - How consistent with the observations $\phi(x_i)$
- Pairwise potential
  - How smooth the predictions are (in spatial or bilateral space)

$$\sum_{x_j \in V_i} \phi(x_i, x_j)$$

CRF Inference

- Belief Propagation
- MCMC
  - Metropolis Hastings
  - Alpha-expansion, alpha-beta swap
- Variational Inference
  - Approximate the energy with a simpler function

- Solving an approximate problem with exact optimization
- Solving an approximate problem with an approximate method

Demo

- Variational mean field approximation
  - Approximate the conditional probability with simpler form
  - Message passing
- https://github.com/chrischoy/segmentation_lecture/blob/master/crf.py
- python crf.py

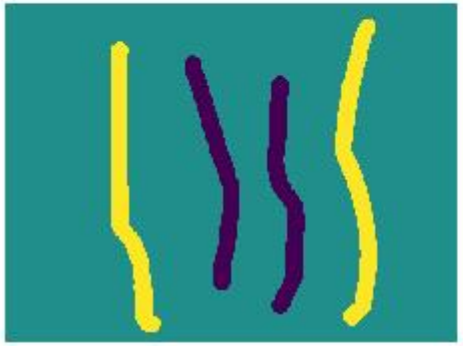
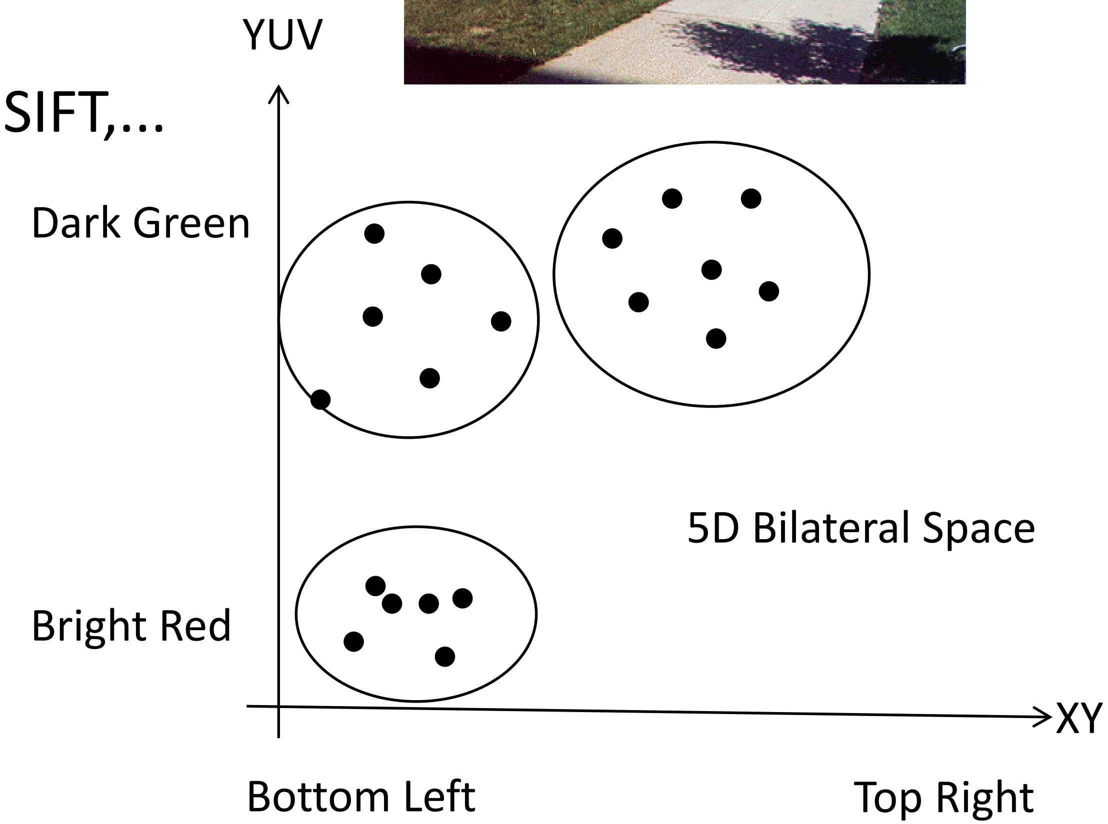
   

Table of Contents

- Types of Segmentations
- Clustering-based Segmentation
  - K-means
  - Mean Shift
- Graph-based Segmentation
  - Normalized Cut, Spectral Clustering
  - Conditional Random Field
- Supervised Segmentation
  - Feature learning
  - Fully Convolutional Neural Network (FCNN)
  - Probabilistic Graphical Model (CRF) + FCNN
  - Spectral Clustering + FCNN
- Example code

Supervised Segmentation

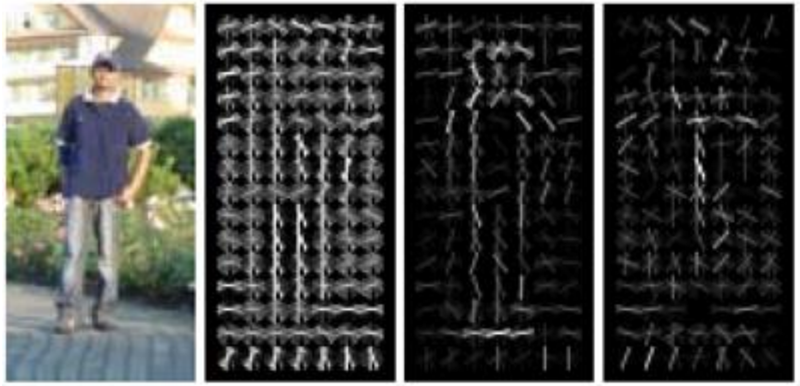
- Image → Feature representation → Classification
- What is a good feature?
  - Bilateral feat, Texton, Bag of words, HOG, SIFT,...
  - Ans: _______
  - Hint: Data Processing Inequality



- Learn
  - Features
  - Parameters in a classifier

Feature Representation

- Image: pixels
  - Pixel or patch
  - Texton, DSIFT, HOG, …: extract a high dimensional feature from a patch
- Dense feature extraction



- Classifier
  - Support Vector Machine
  - Logistic Regression
  - Neural Network
  - …

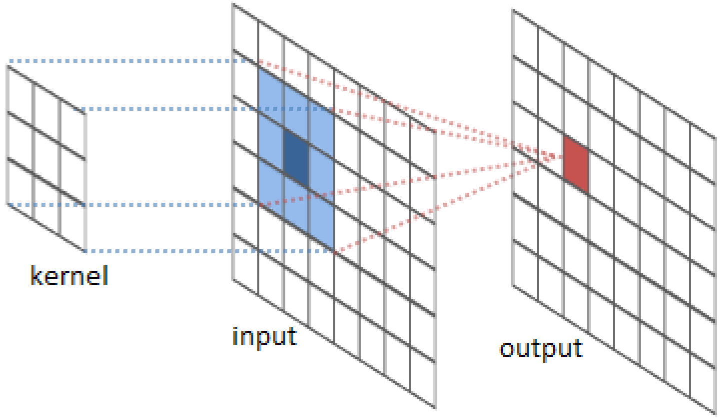
Demo

- Dense feature: 3x3 image patch
- Classifier: SVM
- Optimization: Momentum SGD
  - https://github.com/chrischoy/segmentation_lecture/blob/master/svm.py




Fully Convolutional Neural Network

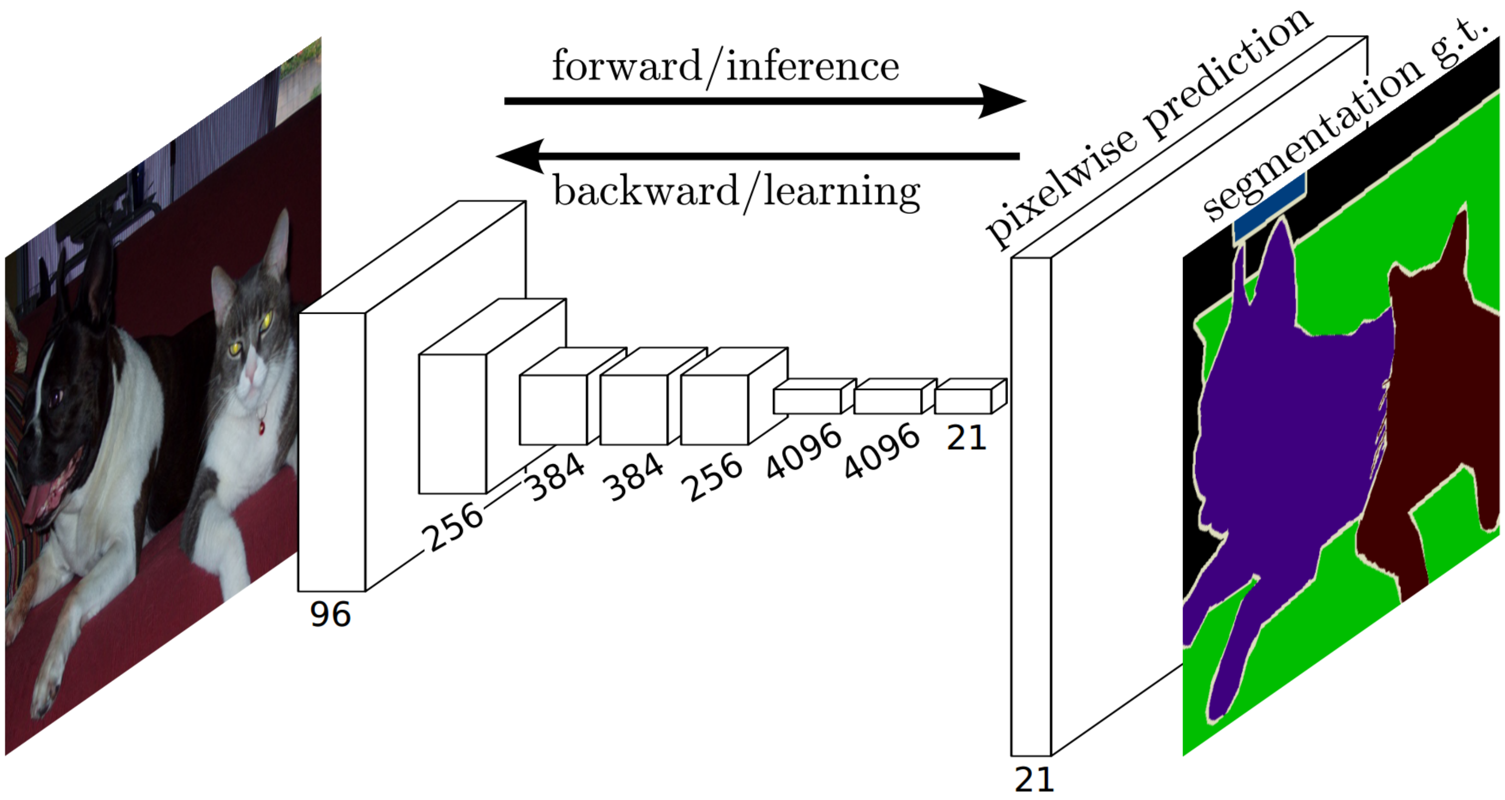
- Demo svm.py
  - 3x3 convolution
- Stack another convolution and non-linearity between them
  - Convolutional Neural Network
- Convolutional Neural networks
  - Function approximators
  - Superclass of all hand designed features
- Fully Convolutional Neural Network



Dense feature extraction using convolutions (No flattening, No fully connected layer)

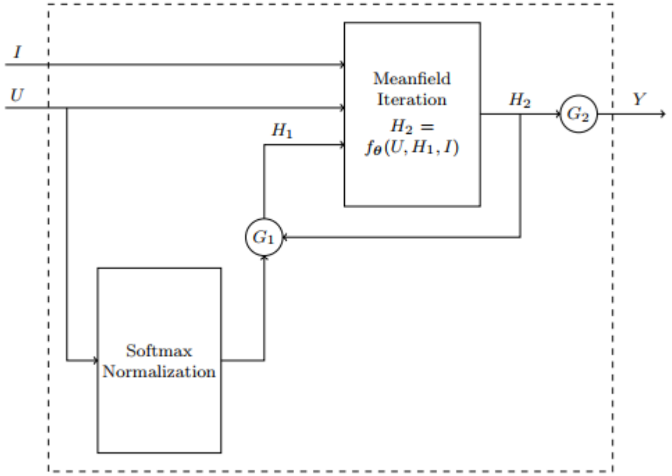
- Loss: Cross-Entropy

Fully Convolutional Neural Network

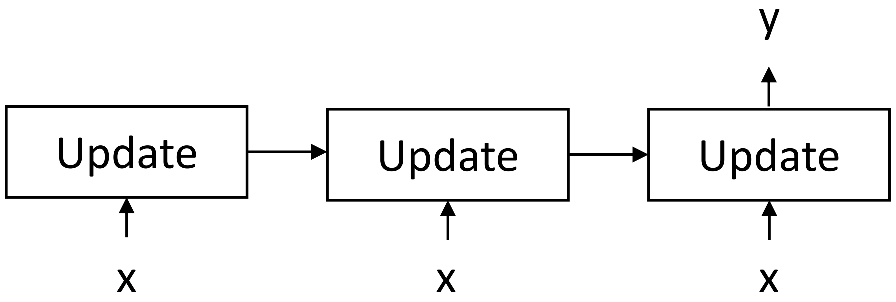


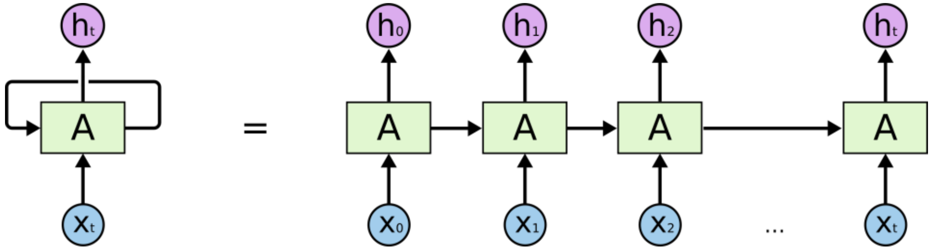
Long, Shelhamer, Darrel (Arxiv)

Neural Network + CRF



- Conditional Random Field
  - Consistency
- Neural Network
  - Strong unary potential
- Differentiable?
  - Initialization, Iterative Approximate Inference
    - Krähenbühl & Koltun, NIPS’11
  - Iterative Inference → Recurrent Neural Network
  - CNN + CRF as RNN

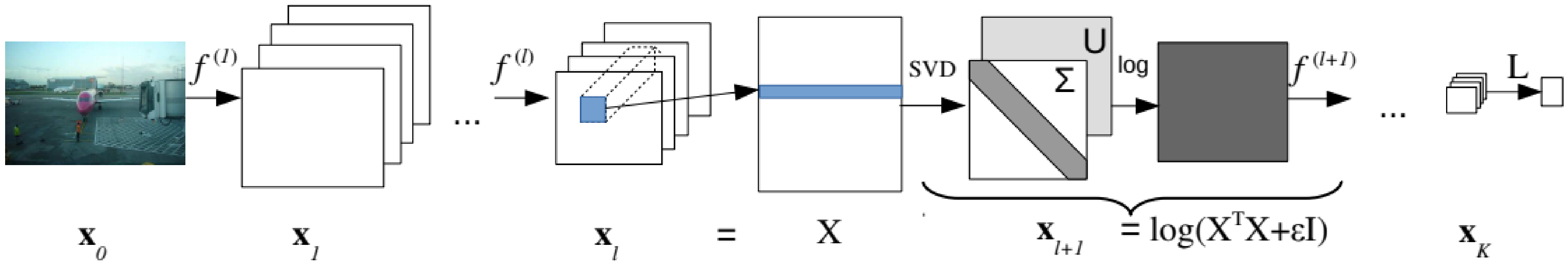




Zheng et al., ICCV ‘15

Neural Network + Spectral Clustering

- Bach & Jordan, Learning Spectral Clustering
  - Continuous relaxation (spectral clustering)
  - Differentiable
- Ionescu et al., Matrix Backpropagation for Deep Networks with Structured Layers
  - Combines neural network for feature extraction



Problem Solving

- Input (feature) representation
- Problem definition
- Approximate optimization
  - Variational Inference
  - Continuous approximation
  - Heuristic optimization
- Learning parameters
  - Gradient, approximate gradient
  - Backpropagation
- Hyper-parameter sweep
  - Validation
- Test