Lecture 10: PGM — Learning structures

Qinfeng (Javen) Shi

15 October 2015

Intro. to Stats. Machine Learning
COMP SCI 4401/7401

Table of Contents I

Overview

Given graph, we can learn the parameters, and do inference.

Overview

Given graph, we can learn the parameters, and do inference.

How to get the graph at the first place?

Overview

Given graph, we can learn the parameters, and do inference.

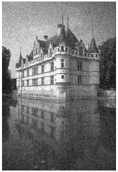
How to get the graph at the first place?

- Manually specify a graph (based on domain knowledge, or experience)
- Use simple rules
- Learn from data
  - Learn from labels (Chow-Liu Tree Algorithm (1968))
  - Learn from both labels and features.

Manually specify a graph

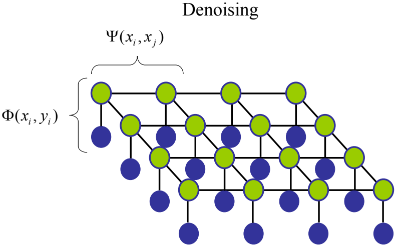
Image denoising[1]

Original



Noisy

Corrected

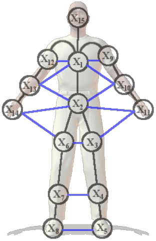


$$X^* = \operatorname{argmax}_X P(X|Y)$$

[1] This example is from Tiberio Caetano's short course: "Machine Learning using Graphical Models"

Manually specify a graph

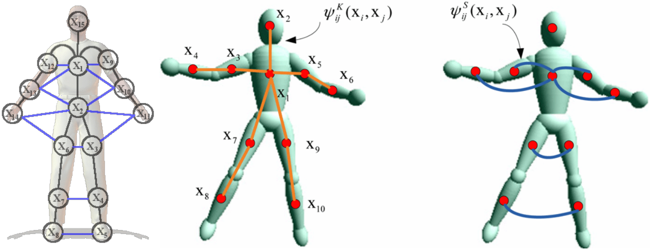
Pose estimation [2]



[2] Left pic from http://cs.brown.edu/~ls/research.html; mid and right pics from http://ieeexplore.ieee.org/ieee_pilot/articles/96jproc10/96jproc10-wu/article.html

Manually specify a graph

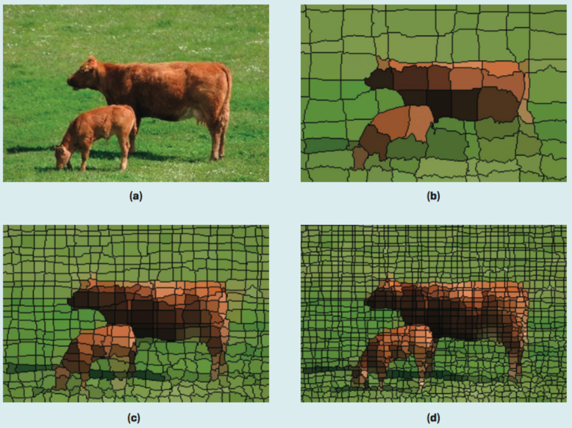
Pose estimation [2]



[2] Left pic from http://cs.brown.edu/~ls/research.html; mid and right pics from http://ieeexplore.ieee.org/ieee_pilot/articles/96jproc10/96jproc10-wu/article.html

Use simple rules

Image segmentation and object recognition (Gould&He, comm. acm 14)

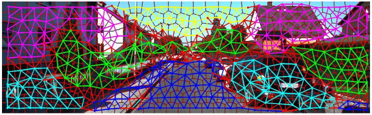


Rules: each small segment (called super-pixel) is a node; if two segments (nodes) are adjacent, add an edge between them.

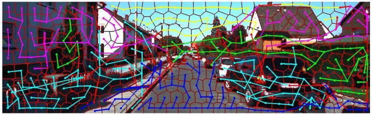
Use simple rules

Scene understanding on the street (driverless cars)

(a) original image



(b) graph via super-pixel **adjacency**

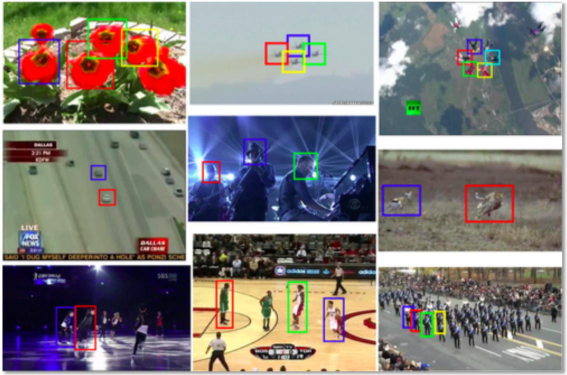


(c) graph via **distance mst (minimal spanning tree)**

Show KITTI dataset if internet works

Use simple rules

Tracking (Zhang&van der Maaten CVPR13)



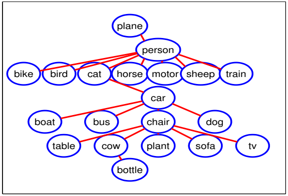
Rule: each bounding box is a node, and find distance mst (minimal spanning tree) among all nodes.

Show their demo if internet works

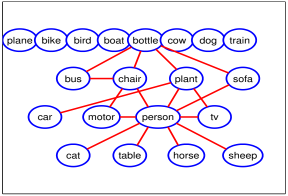
Learn from labels

Multiple labels classification (Tan etal CVPR15)

(d) Images of “plane” in PASCAL2007 database



(e) Learn from labels



(f) Learn from labels and features

Figure : Comparison of graphs learned from PASCAL2007

Why learning from data possible?

Why learning from data possible?

- PGM represents the distribution, and you can estimate the distribution

Why learning from data possible?

- PGM represents the distribution, and you can estimate the distribution
- PGM carries independencies and dependencies, and you can estimate them too

Break

Take a break ...

Chow-Liu Tree Algorithm

The goal: to find a tree structure[3] PGM whose distribution is closest to the underlying distribution.

- learn from labels only
- proposed for Bayes Net initially (directed edges) in 1968
- works for MRFs too with slight modification (undirected edges)

[3] Not for all trees: for Bayes Net, each node can have at most 1 parent.

Mutual info

Definition (Mutual information)

$$I(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log\left(\frac{P(x_i, x_j)}{P(x_i)P(x_j)}\right)$$

Properties:

- $I(X_i, X_j) \geq 0$.
- $I(X_i, X_j) = 0$ if and only if X_i, X_j are independent. (prove it in Assignment 3).

Mutual info

Definition (Mutual information)

$$I(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \left(\frac{P(x_i, x_j)}{P(x_i)P(x_j)} \right)$$

Properties:

- $I(X_i, X_j) \geq 0$.
- $I(X_i, X_j) = 0$ if and only if X_i, X_j are independent. (prove it in Assignment 3).

Intuition: the higher $I(X_i, X_j)$ is, the more correlated X_i, X_j are.

KL divergence

Definition (Kullback-Leibler divergence)

For any distributions $P(\mathbf{x})$ and $P'(\mathbf{x})$ over $\mathbf{x}$,

$$KL(P(\mathbf{x})||P'(\mathbf{x})) = \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{P'(\mathbf{x})}$$

Properties:

- $KL(P(\mathbf{x})||P'(\mathbf{x})) \geq 0$.
- $KL(P(\mathbf{x})||P'(\mathbf{x})) = 0$ if and only if $P(\mathbf{x}), P'(\mathbf{x})$ are the same. (prove it in Assignment 3).

KL divergence

Definition (Kullback-Leibler divergence)

For any distributions $P(\mathbf{x})$ and $P'(\mathbf{x})$ over $\mathbf{x}$,

$$KL(P(\mathbf{x})||P'(\mathbf{x})) = \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{P'(\mathbf{x})}$$

Properties:

- $KL(P(\mathbf{x})||P'(\mathbf{x})) \geq 0$.
- $KL(P(\mathbf{x})||P'(\mathbf{x})) = 0$ if and only if $P(\mathbf{x}), P'(\mathbf{x})$ are the same. (prove it in Assignment 3).

Intuition: the smaller $KL(P(\mathbf{x})||P'(\mathbf{x}))$ is, the closer $P(\mathbf{x}), P'(\mathbf{x})$ are.

Minimise KL divergence

Setting: Let $P(\mathbf{x})$ be the joint distribution of n discrete variables $x_1, x_2, \ldots, x_n$, where $\mathbf{x}$ denotes $(x_1, x_2, \ldots, x_n)$.

Goal: For Bayes Net, we seek a tree structure, whose $P_t(\mathbf{x})$ is closest to $P(\mathbf{x})$ in the sense of smallest $KL(P(\mathbf{x})||P_t(\mathbf{x}))$ with one condition: each node in the tree t has at most 1 parent. (show some examples in the document camera)

$$\min KL(P(\mathbf{x})||P_t(\mathbf{x}))$$

Bayes Net: $P_t(\mathbf{x}) = \prod_{i=1}^{n} P(x_i|pa_t(x_i))$.
Note: $pa_t(x_i)$ (*i.e.* parent of x_i) is encoded in the tree t.

Minimise KL divergence

$$KL(P(\mathbf{x})||P_t(\mathbf{x})) = \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{P_t(\mathbf{x})}$$

$$= -\sum_{\mathbf{x}} P(\mathbf{x}) \log P_t(\mathbf{x}) + \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x})$$

$$= -\sum_{\mathbf{x}} P(\mathbf{x}) \log \left(\prod_{i=1}^{n} P(x_i|pa_t(x_i)) \right) + \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x})$$

$$= -\sum_{\mathbf{x}} P(\mathbf{x}) \sum_{i=1}^{n} \log \left(\frac{P(x_i, pa_t(x_i))}{P(pa_t(x_i))} \right) + \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x})$$

$$= -\sum_{\mathbf{x}} P(\mathbf{x}) \sum_{i=1}^{n} \log \left(\frac{P(x_i, pa_t(x_i))P(x_i)}{P(x_i)P(pa_t(x_i))} \right) + \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x})$$

$$= -\sum_{\mathbf{x}} P(\mathbf{x}) \sum_{i=1}^{n} \log \left(\frac{P(x_i, pa_t(x_i))}{P(x_i)P(pa_t(x_i))} \right) \underbrace{- \sum_{\mathbf{x}} P(\mathbf{x}) \sum_{i=1}^{n} \log P(x_i)}_{:= const_1} + \underbrace{\sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x})}_{:= const_2}$$

Minimise KL divergence

$$= -\sum_{\mathbf{x}} P(\mathbf{x}) \sum_{i=1}^{n} \log\left(\frac{P(x_i, pa_t(x_i))}{P(x_i)P(pa_t(x_i))}\right) - \underbrace{\sum_{\mathbf{x}} P(\mathbf{x}) \sum_{i=1}^{n} \log P(x_i)}_{:=const_1} + \underbrace{\sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x})}_{:=const_2}$$

$$= -\sum_{i=1}^{n} \left[\sum_{\mathbf{x}} P(\mathbf{x}) \log\left(\frac{P(x_i, pa_t(x_i))}{P(x_i)P(pa_t(x_i))}\right)\right] - const_1 + const_2$$

$$= -\sum_{i=1}^{n} \underbrace{\left[\sum_{x_i, pa_t(x_i)} P(x_i, pa_t(x_i)) \log\left(\frac{P(x_i, pa_t(x_i))}{P(x_i)P(pa_t(x_i))}\right)\right]}_{I(X_i, pa_t(X_i))} - const_1 + const_2$$

$$= -\sum_{i=1}^{n} I(X_i, pa_t(X_i)) - const_1 + const_2$$

So to minimise KL div is to maximise mutual info,

$$\operatorname*{argmin}_{t \in \mathcal{T}} KL(P(\mathbf{x}) || P_t(\mathbf{x})) = \operatorname*{argmax}_{t \in \mathcal{T}} \sum_{i=1}^{n} I(X_i, pa_t(X_i)).$$

Algorithm

Steps (both Bayes Nets and MRFs):

1. compute all pairwise $I(X_i, X_j)$
2. find the minimal spanning tree w.r.t. mutual info
3. decide the direction of the edges for Bayes Nets (this step is not needed for MRFs)

Algorithm

Steps (both Bayes Nets and MRFs):

1. compute all pairwise $I(X_i, X_j)$
2. find the minimal spanning tree w.r.t. mutual info
3. decide the direction of the edges for Bayes Nets (this step is not needed for MRFs)

How to do step 2?

Algorithm

Steps (both Bayes Nets and MRFs):

1. compute all pairwise $I(X_i, X_j)$
2. find the minimal spanning tree w.r.t. mutual info
3. decide the direction of the edges for Bayes Nets (this step is not needed for MRFs)

How to do step 2?

Sort $I(X_i, X_j)$ from highest to lowest. Keep adding edges in that order, and skip if adding it would cause a loop (i.e. not a tree any more).

Algorithm

Steps (both Bayes Nets and MRFs):

1. compute all pairwise $I(X_i, X_j)$
2. find the minimal spanning tree w.r.t. mutual info
3. decide the direction of the edges for Bayes Nets (this step is not needed for MRFs)

How to do step 2?

Sort $I(X_i, X_j)$ from highest to lowest. Keep adding edges in that order, and skip if adding it would cause a loop (i.e. not a tree any more).

How many edges to add?

Algorithm

Steps (both Bayes Nets and MRFs):

1. compute all pairwise $I(X_i, X_j)$
2. find the minimal spanning tree w.r.t. mutual info
3. decide the direction of the edges for Bayes Nets (this step is not needed for MRFs)

How to do step 2?

Sort $I(X_i, X_j)$ from highest to lowest. Keep adding edges in that order, and skip if adding it would cause a loop (i.e. not a tree any more).

How many edges to add?

$n - 1$ many edges (for n many nodes/variables).

Algorithm

Steps (both Bayes Nets and MRFs):

1. compute all pairwise $I(X_i, X_j)$
2. find the minimal spanning tree w.r.t. mutual info
3. decide the direction of the edges for Bayes Nets (this step is not needed for MRFs)

How to do step 2?

Sort $I(X_i, X_j)$ from highest to lowest. Keep adding edges in that order, and skip if adding it would cause a loop (i.e. not a tree any more).

How many edges to add?

$n - 1$ many edges (for n many nodes/variables).

Example in document camera

That's all

Thanks!