Part 2: Introduction to Graphical Models

Sebastian Nowozin and Christoph H. Lampert

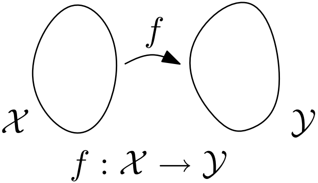
Colorado Springs, 25th June 2011

Microsoft Research

IST AUSTRIA Institute of Science and Technology

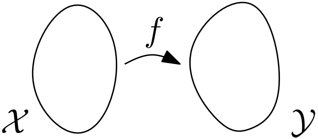
Introduction

- Model: relating observations x to quantities of interest y
- Example 1: given RGB image x, infer depth y for each pixel
- Example 2: given RGB image x, infer presence and positions y of all objects shown

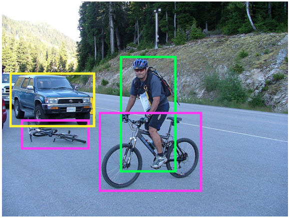


Introduction

- Model: relating observations x to quantities of interest y
- Example 1: given RGB image x, infer depth y for each pixel
- Example 2: given RGB image x, infer presence and positions y of all objects shown



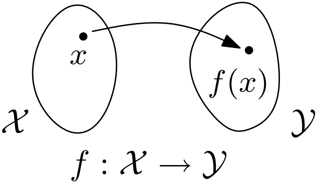
$f : \mathcal{X} \to \mathcal{Y}$



$\mathcal{X}$: image, $\mathcal{Y}$: object annotations

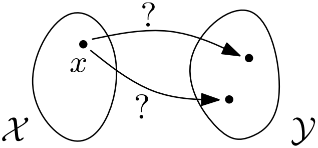
Introduction

- General case: mapping $x \in \mathcal{X}$ to $y \in \mathcal{Y}$
- Graphical models are a concise *language* to define this mapping
- Mapping can be *ambiguous*: measurement noise, lack of well-posedness (e.g. occlusions)
- Probabilistic graphical models: define form $p(y|x)$ or $p(x, y)$ for all $y \in \mathcal{Y}$



Introduction

- General case: mapping $x \in \mathcal{X}$ to $y \in \mathcal{Y}$
- Graphical models are a concise *language* to define this mapping
- Mapping can be *ambiguous*: measurement noise, lack of well-posedness (e.g. occlusions)
- Probabilistic graphical models: define form $p(y|x)$ or $p(x, y)$ for all $y \in \mathcal{Y}$



$$p(Y|X = x)$$

Graphical Models

A graphical model defines

- a *family of probability distributions* over a set of random variables,
- by means of a graph,
- so that the random variables satisfy *conditional independence assumptions* encoded in the graph.

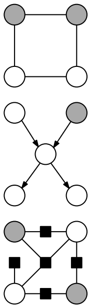
Graphical Models

A graphical model defines

- a *family of probability distributions* over a set of random variables,
- by means of a graph,
- so that the random variables satisfy *conditional independence assumptions* encoded in the graph.

Popular classes of graphical models,

- Undirected graphical models (Markov random fields),
- Directed graphical models (Bayesian networks),
- Factor graphs,
- Others: chain graphs, influence diagrams, etc.



Bayesian Networks

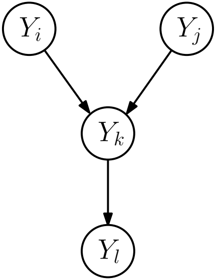
- Graph: $G = (V, \mathcal{E})$, $\mathcal{E} \subset V \times V$
  - directed
  - acyclic
- Variable domains $\mathcal{Y}_i$
- Factorization

$$p(Y = y) = \prod_{i \in V} p(y_i | y_{\mathrm{pa}_G(i)})$$

over distributions, by conditioning on parent nodes.

- Example

$$p(Y = y) = p(Y_l = y_l | Y_k = y_k) p(Y_k = y_k | Y_i = y_i, Y_j = y_j) p(Y_i = y_i) p(Y_j = y_j).$$



A simple Bayes net

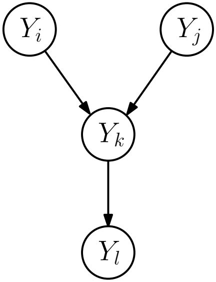
Bayesian Networks

- Graph: $G = (V, \mathcal{E})$, $\mathcal{E} \subset V \times V$
  - directed
  - acyclic
- Variable domains $\mathcal{Y}_i$
- Factorization

$$p(Y = y) = \prod_{i \in V} p(y_i | y_{\mathrm{pa}_G(i)})$$

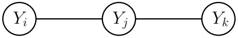
  over distributions, by conditioning on parent nodes.
- Example

$$p(Y = y) = p(Y_l = y_l | Y_k = y_k) p(Y_k = y_k | Y_i = y_i, Y_j = y_j) \\ p(Y_i = y_i) p(Y_j = y_j).$$



A simple Bayes net

Undirected Graphical Models



A simple MRF

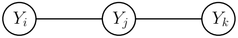
- = Markov random field (MRF) = Markov network
- Graph: $G = (V, \mathcal{E})$, $\mathcal{E} \subset V \times V$
  - undirected, no self-edges
- Variable domains $\mathcal{Y}_i$
- Factorization over potentials ψ at *cliques*,

$$p(y) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(y_C)$$

- Constant $Z = \sum_{y \in \mathcal{Y}} \prod_{C \in \mathcal{C}(G)} \psi_C(y_C)$
- Example

$$p(y) = \frac{1}{Z} \psi_i(y_i) \psi_j(y_j) \psi_l(y_l) \psi_{i,j}(y_i, y_j)$$

Undirected Graphical Models



A simple MRF

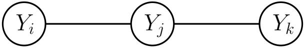
- = Markov random field (MRF) = Markov network
- Graph: $G = (V, \mathcal{E})$, $\mathcal{E} \subset V \times V$
  - undirected, no self-edges
- Variable domains $\mathcal{Y}_i$
- Factorization over potentials ψ at *cliques*,

$$p(y) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(y_C)$$

- Constant $Z = \sum_{y \in \mathcal{Y}} \prod_{C \in \mathcal{C}(G)} \psi_C(y_C)$
- Example

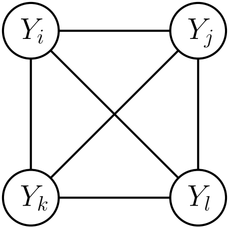
$$p(y) = \frac{1}{Z} \psi_i(y_i) \psi_j(y_j) \psi_l(y_l) \psi_{i,j}(y_i, y_j)$$

Example 1



- *Cliques* $\mathcal{C}(G)$: set of vertex sets V' with $V' \subseteq V$, $\mathcal{E} \cap (V' \times V') = V' \times V'$
- Here $\mathcal{C}(G) = \{\{i\}, \{i,j\}, \{j\}, \{j,k\}, \{k\}\}$
- $$p(y) = \frac{1}{Z} \psi_i(y_i) \psi_j(y_j) \psi_l(y_l) \psi_{i,j}(y_i, y_j)$$

Example 2



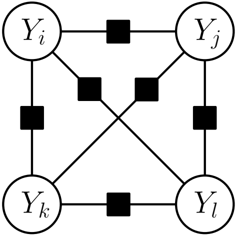
- Here $\mathcal{C}(G) = 2^V$: all subsets of V are cliques
- $$p(y) = \frac{1}{Z} \prod_{A \in 2^{\{i,j,k,l\}}} \psi_A(y_A).$$

Factor Graphs

- Graph: $G = (V, \mathcal{F}, \mathcal{E})$, $\mathcal{E} \subseteq V \times \mathcal{F}$
  - variable nodes V,
  - factor nodes $\mathcal{F}$,
  - edges $\mathcal{E}$ between variable and factor nodes.
  - *scope* of a factor, $N(F) = \{i \in V : (i, F) \in \mathcal{E}\}$
- Variable domains $\mathcal{Y}_i$
- Factorization over potentials ψ at *factors*,

$$p(y) = \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)})$$

- Constant $Z = \sum_{y \in \mathcal{Y}} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)})$



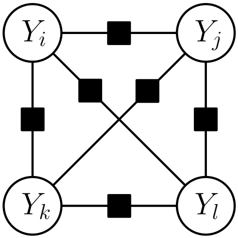
Factor graph

Factor Graphs

- Graph: $G = (V, \mathcal{F}, \mathcal{E})$, $\mathcal{E} \subseteq V \times \mathcal{F}$
  - variable nodes V,
  - factor nodes $\mathcal{F}$,
  - edges $\mathcal{E}$ between variable and factor nodes.
  - *scope* of a factor,
    $N(F) = \{i \in V : (i, F) \in \mathcal{E}\}$
- Variable domains $\mathcal{Y}_i$
- Factorization over potentials ψ at *factors*,

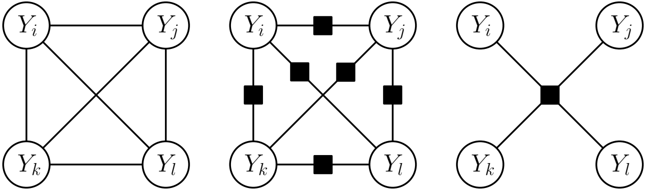
$$p(y) = \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)})$$

- Constant $Z = \sum_{y \in \mathcal{Y}} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)})$



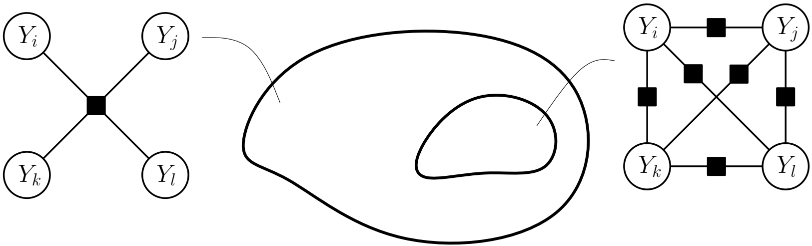
Factor graph

Why factor graphs?



- Factor graphs are *explicit* about the factorization
- Hence, easier to work with
- Universal (just like MRFs and Bayesian networks)

Capacity



- Factor graph defines family of distributions
- Some families are larger than others

Four remaining pieces

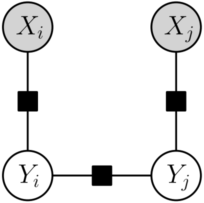
1. Conditional distributions (CRFs)
2. Parameterization
3. Test-time inference
4. Learning the model from training data

Four remaining pieces

1. Conditional distributions (CRFs)
2. Parameterization
3. Test-time inference
4. Learning the model from training data

Conditional Distributions

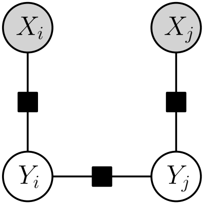
- We have discussed $p(y)$,
- How do we define $p(y|x)$?
- Potentials become a function of $x_{N(F)}$
- Partition function depends on x
- Conditional random fields (CRFs)
- x is not part of the probability model, i.e. not treated as random variable



conditional distribution

Conditional Distributions

- We have discussed $p(y)$,
- How do we define $p(y|x)$?
- Potentials become a function of $x_{N(F)}$
- Partition function depends on x
- Conditional random fields (CRFs)
- x is not part of the probability model, i.e. not treated as random variable



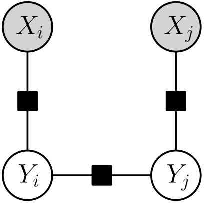
conditional distribution

$$p(y) = \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)})$$

$$p(y|x) = \frac{1}{Z(x)} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)}; x_{N(F)})$$

Conditional Distributions

- We have discussed $p(y)$,
- How do we define $p(y|x)$?
- Potentials become a function of $x_{N(F)}$
- Partition function depends on x
- Conditional random fields (CRFs)
- x is not part of the probability model, i.e. not treated as random variable



conditional distribution

$$p(y) = \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)})$$

$$p(y|x) = \frac{1}{Z(x)} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)}; x_{N(F)})$$

Potentials and Energy Functions

- For each factor $F \in \mathcal{F}$, $\mathcal{Y}_F = \bigtimes_{i \in N(F)} \mathcal{Y}_i$,

$$E_F : \mathcal{Y}_{N(F)} \to \mathbb{R},$$

- Potentials and energies (assume $\psi_F(y_F) > 0$)

$$\psi_F(y_F) = \exp(-E_F(y_F)), \quad \text{and} \quad E_F(y_F) = -\log(\psi_F(y_F)).$$

- Then $p(y)$ can be written as

$$\begin{aligned} p(Y = y) &= \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(y_F) \\ &= \frac{1}{Z} \exp(-\sum_{F \in \mathcal{F}} E_F(y_F)), \end{aligned}$$

- Hence, $p(y)$ is completely determined by $E(y) = \sum_{F \in \mathcal{F}} E_F(y_F)$

Potentials and Energy Functions

- For each factor $F \in \mathcal{F}$, $\mathcal{Y}_F = \bigtimes_{i \in N(F)} \mathcal{Y}_i$,

$$E_F : \mathcal{Y}_{N(F)} \rightarrow \mathbb{R},$$

- Potentials and energies (assume $\psi_F(y_F) > 0$)

$$\psi_F(y_F) = \exp(-E_F(y_F)), \quad \text{and} \quad E_F(y_F) = -\log(\psi_F(y_F)).$$

- Then $p(y)$ can be written as

$$\begin{aligned} p(Y = y) &= \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(y_F) \\ &= \frac{1}{Z} \exp(-\sum_{F \in \mathcal{F}} E_F(y_F)), \end{aligned}$$

- Hence, $p(y)$ is completely determined by $E(y) = \sum_{F \in \mathcal{F}} E_F(y_F)$

Potentials and Energy Functions

- For each factor $F \in \mathcal{F}$, $\mathcal{Y}_F = \bigtimes_{i \in N(F)} \mathcal{Y}_i$,

$$E_F : \mathcal{Y}_{N(F)} \to \mathbb{R},$$

- Potentials and energies (assume $\psi_F(y_F) > 0$)

$$\psi_F(y_F) = \exp(-E_F(y_F)), \quad \text{and} \quad E_F(y_F) = -\log(\psi_F(y_F)).$$

- Then $p(y)$ can be written as

$$\begin{aligned} p(Y = y) &= \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(y_F) \\ &= \frac{1}{Z} \exp(-\sum_{F \in \mathcal{F}} E_F(y_F)), \end{aligned}$$

- Hence, $p(y)$ is completely determined by $E(y) = \sum_{F \in \mathcal{F}} E_F(y_F)$

Energy Minimization

$$
\begin{aligned}
\operatorname*{argmax}_{y \in \mathcal{Y}} p(Y = y) &= \operatorname*{argmax}_{y \in \mathcal{Y}} \frac{1}{Z} \exp(-\sum_{F \in \mathcal{F}} E_F(y_F)) \\
&= \operatorname*{argmax}_{y \in \mathcal{Y}} \exp(-\sum_{F \in \mathcal{F}} E_F(y_F)) \\
&= \operatorname*{argmax}_{y \in \mathcal{Y}} -\sum_{F \in \mathcal{F}} E_F(y_F) \\
&= \operatorname*{argmin}_{y \in \mathcal{Y}} \sum_{F \in \mathcal{F}} E_F(y_F) \\
&= \operatorname*{argmin}_{y \in \mathcal{Y}} E(y).
\end{aligned}
$$

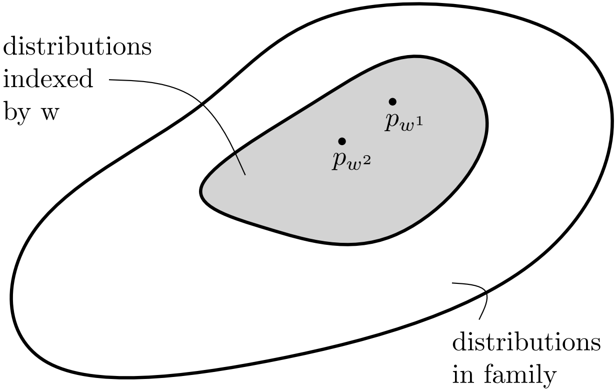
- Energy minimization can be interpreted as solving for the most likely state of some factor graph model

Parameterization

- Factor graphs define a family of distributions
- *Parameterization*: identifying individual members by parameters w

Parameterization

- Factor graphs define a family of distributions
- *Parameterization*: identifying individual members by parameters w



Example: Parameterization

- Image segmentation model
- Pairwise "Potts" energy function $E_F(y_i, y_j; w_1)$,

$$E_F : \{0,1\} \times \{0,1\} \times \mathbb{R} \to \mathbb{R},$$

- $E_F(0,0;w_1) = E_F(1,1;w_1) = 0$
- $E_F(0,1;w_1) = E_F(1,0;w_1) = w_1$

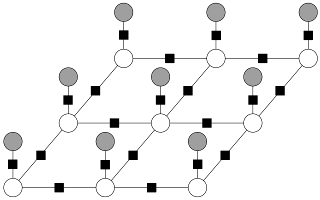


image segmentation model

Example: Parameterization (cont)

- Image segmentation model
- Unary energy function $E_F(y_i; x, w)$,

$$E_F : \{0,1\} \times \mathcal{X} \times \mathbb{R}^{\{0,1\} \times D} \to \mathbb{R},$$

- $E_F(0; x, w) = \langle w(0), \psi_F(x) \rangle$
- $E_F(1; x, w) = \langle w(1), \psi_F(x) \rangle$
- Features $\psi_F : \mathcal{X} \to \mathbb{R}^D$, e.g. image filters

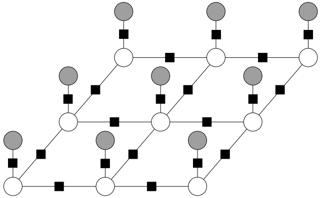
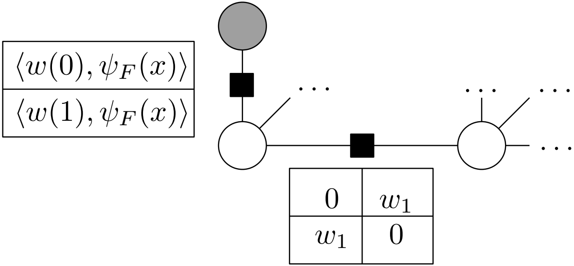
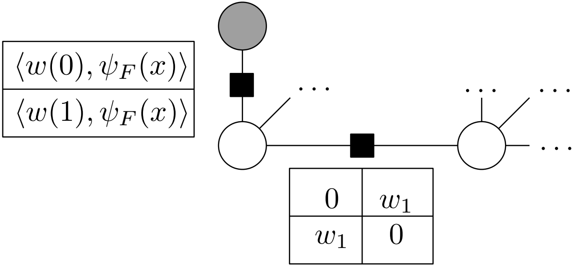


image segmentation model

Example: Parameterization (cont)



Example: Parameterization (cont)



- Total number of parameters: $D + D + 1$
- Parameters are *shared*, but energies differ because of different $\psi_F(x)$
- General form, linear in w,

$$E_F(y_F; x_F, w) = \langle w(y_F), \psi_F(x_F) \rangle$$

Making Predictions

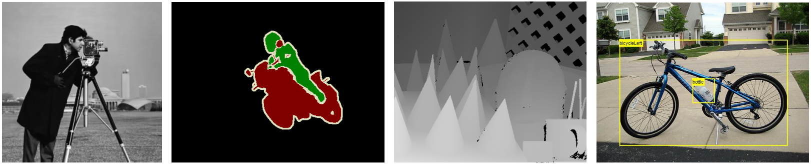
- Making predictions: given $x \in \mathcal{X}$, predict $y \in \mathcal{Y}$
- How to measure quality of prediction? (or function $f : \mathcal{X} \to \mathcal{Y}$)

Loss function

- Define a *loss function*

$$\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+,$$

  so that $\Delta(y, y^*)$ measures the loss incurred by predicting y when y^* is true.
- The *loss function* is application dependent



Test-time Inference

- *Loss function* $\Delta(y, f(x))$: correct label y, predict $f(x)$

$$\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

- True joint distribution $d(X, Y)$ and true conditional $d(y|x)$
- Model distribution $p(y|x)$
- *Expected loss*: quality of prediction

$$\begin{aligned} \mathcal{R}_f^{\Delta}(x) &= \mathbb{E}_{y \sim d(y|x)} \Delta(y, f(x)) \\ &= \sum_{y \in \mathcal{Y}} d(y|x)\, \Delta(y, f(x)). \\ &\approx \mathbb{E}_{y \sim p(y|x;w)} \Delta(y, f(x)) \end{aligned}$$

- Assuming that $p(y|x; w) \approx d(y|x)$

Test-time Inference

- *Loss function* $\Delta(y, f(x))$: correct label y, predict $f(x)$

$$\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

- True joint distribution $d(X, Y)$ and true conditional $d(y|x)$
- Model distribution $p(y|x)$
- *Expected loss*: quality of prediction

$$\begin{aligned} \mathcal{R}_f^{\Delta}(x) &= \mathbb{E}_{y \sim d(y|x)} \Delta(y, f(x)) \\ &= \sum_{y \in \mathcal{Y}} d(y|x) \Delta(y, f(x)). \\ &\approx \mathbb{E}_{y \sim p(y|x;w)} \Delta(y, f(x)) \end{aligned}$$

- Assuming that $p(y|x;w) \approx d(y|x)$

Test-time Inference

- *Loss function* $\Delta(y, f(x))$: correct label y, predict $f(x)$

$$\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

- True joint distribution $d(X, Y)$ and true conditional $d(y|x)$
- Model distribution $p(y|x)$
- *Expected loss*: quality of prediction

$$\begin{aligned} \mathcal{R}_f^{\Delta}(x) &= \mathbb{E}_{y \sim d(y|x)} \Delta(y, f(x)) \\ &= \sum_{y \in \mathcal{Y}} d(y|x) \Delta(y, f(x)). \\ &\approx \mathbb{E}_{y \sim p(y|x;w)} \Delta(y, f(x)) \end{aligned}$$

- Assuming that $p(y|x; w) \approx d(y|x)$

Example 1: 0/1 loss

Loss 0 iff perfectly predicted, 1 otherwise:

$$\Delta_{0/1}(y, y^*) = I(y \neq y^*) = \begin{cases} 0 & \text{if } y = y^* \\ 1 & \text{otherwise} \end{cases}$$

Plugging it in,

$$\begin{aligned} y^* &:= \operatorname*{argmin}_{y' \in \mathcal{Y}} \mathbb{E}_{y \sim p(y|x)} \left[\Delta_{0/1}(y, y')\right] \\ &= \operatorname*{argmax}_{y' \in \mathcal{Y}} p(y'|x) \\ &= \operatorname*{argmin}_{y' \in \mathcal{Y}} E(y', x). \end{aligned}$$

- Minimizing the expected 0/1-loss → MAP prediction (energy minimization)

Example 1: 0/1 loss

Loss 0 iff perfectly predicted, 1 otherwise:

$$\Delta_{0/1}(y, y^*) = I(y \neq y^*) = \begin{cases} 0 & \text{if } y = y^* \\ 1 & \text{otherwise} \end{cases}$$

Plugging it in,

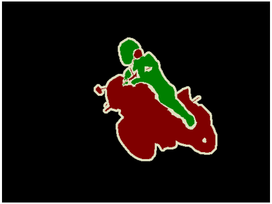
$$\begin{aligned} y^* \quad &:= \quad \operatorname*{argmin}_{y' \in \mathcal{Y}} \mathbb{E}_{y \sim p(y|x)} \left[\Delta_{0/1}(y, y')\right] \\ &= \quad \operatorname*{argmax}_{y' \in \mathcal{Y}} p(y'|x) \\ &= \quad \operatorname*{argmin}_{y' \in \mathcal{Y}} E(y', x). \end{aligned}$$

- Minimizing the expected 0/1-loss → MAP prediction (energy minimization)

Example 2: Hamming loss

Count the number of mislabeled variables:

$$\Delta_H(y, y^*) = \frac{1}{|V|} \sum_{i \in V} I(y_i \neq y_i^*)$$



Plugging it in,

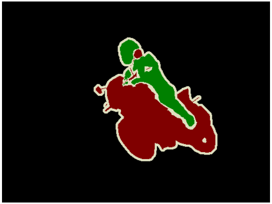
$$\begin{aligned} y^* &:= \operatorname*{argmin}_{y' \in \mathcal{Y}} \mathbb{E}_{y \sim p(y|x)} \left[\Delta_H(y, y')\right] \\ &= \left(\operatorname*{argmax}_{y_i' \in \mathcal{Y}_i} p(y_i'|x) \right)_{i \in V} \end{aligned}$$

- Minimizing the expected Hamming loss → *maximum posterior marginal* (MPM, Max-Marg) prediction

Example 2: Hamming loss

Count the number of mislabeled variables:

$$\Delta_H(y, y^*) = \frac{1}{|V|} \sum_{i \in V} I(y_i \neq y_i^*)$$



Plugging it in,

$$y^* := \operatorname*{argmin}_{y' \in \mathcal{Y}} \mathbb{E}_{y \sim p(y|x)} \left[\Delta_H(y, y')\right]$$

$$= \left(\operatorname*{argmax}_{y_i' \in \mathcal{Y}_i} p(y_i'|x) \right)_{i \in V}$$

- Minimizing the expected Hamming loss → *maximum posterior marginal* (MPM, Max-Marg) prediction

Example 3: Squared error

Assume a vector space on $\mathcal{Y}_i$ (pixel intensities, optical flow vectors, etc.).
Sum of squared errors

$$\Delta_Q(y, y^*) = \frac{1}{|V|} \sum_{i \in V} \|y_i - y_i^*\|^2.$$

Plugging it in,

$$\begin{aligned} y^* &:= \operatorname*{argmin}_{y' \in \mathcal{Y}} \mathbb{E}_{y \sim p(y|x)} [\Delta_Q(y, y')] \\ &= \left(\sum_{y_i' \in \mathcal{Y}_i} p(y_i'|x) y_i' \right)_{i \in V} \end{aligned}$$

- Minimizing the expected squared error → *minimum mean squared error* (MMSE) prediction

Example 3: Squared error

Assume a vector space on $\mathcal{Y}_i$ (pixel intensities, optical flow vectors, etc.).
Sum of squared errors

$$\Delta_Q(y, y^*) = \frac{1}{|V|} \sum_{i \in V} \|y_i - y_i^*\|^2.$$

Plugging it in,

$$y^* := \operatorname*{argmin}_{y' \in \mathcal{Y}} \mathbb{E}_{y \sim p(y|x)} [\Delta_Q(y, y')]$$

$$= \left(\sum_{y_i' \in \mathcal{Y}_i} p(y_i'|x) y_i' \right)_{i \in V}$$

- Minimizing the expected squared error → *minimum mean squared error* (MMSE) prediction

Inference Task: Maximum A Posteriori (MAP) Inference

Definition (Maximum A Posteriori (MAP) Inference)

Given a factor graph, parameterization, and weight vector w, and given the observation x, find

$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}} p(Y = y | x, w) = \operatorname*{argmin}_{y \in \mathcal{Y}} E(y; x, w).$$

Inference Task: Probabilistic Inference

Definition (Probabilistic Inference)

Given a factor graph, parameterization, and weight vector w, and given the observation x, find

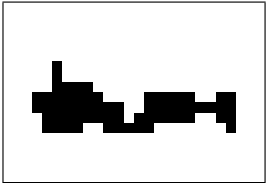
$$\log Z(x,w) = \log \sum_{y \in \mathcal{Y}} \exp(-E(y;x,w)),$$

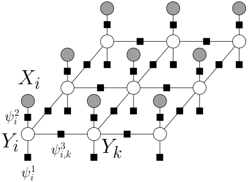
$$\mu_F(y_F) = p(Y_F = y_f | x, w), \quad \forall F \in \mathcal{F}, \forall y_F \in \mathcal{Y}_F.$$

- This typically includes variable marginals

$$\mu_i(y_i) = p(y_i | x, w)$$

Example: Man-made structure detection

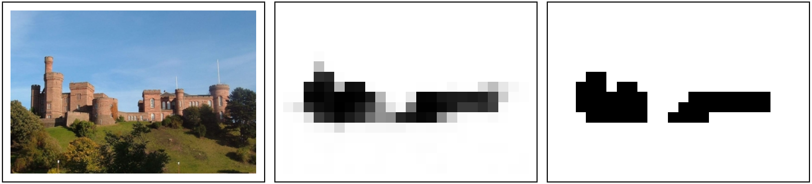




- Left: input image x,
- Middle: ground truth labeling on 16-by-16 pixel blocks,
- Right: factor graph model

- Features: gradient and color histograms
- Estimate model parameters from $\approx$ 60 training images

Example: Man-made structure detection



- Left: input image x,
- Middle (probabilistic inference): visualization of the variable marginals $p(y_i = \text{"}manmade\text{"} | x, w)$,
- Right (MAP inference): joint MAP labeling $y^* = \operatorname{argmax}_{y \in \mathcal{Y}} p(y|x, w)$.

Training the Model

What can be learned?

- Model structure: factors
- Model variables: observed variables fixed, but we can add unobserved variables
- Factor energies: parameters

Training the Model

What can be learned?

- Model structure: factors
- Model variables: observed variables fixed, but we can add unobserved variables
- Factor energies: parameters

Training: Overview

- Assume a fully observed, independent and identically distributed (iid) sample set

$$\{(x^n, y^n)\}_{n=1,\dots,N}, \qquad (x^n, y^n) \sim d(X, Y)$$

- Goal: predict well,
- Alternative goal: first model $d(y|x)$ well by $p(y|x, w)$, then predict by minimizing the expected loss

Probabilistic Learning

Problem (Probabilistic Parameter Learning)

Let $d(y|x)$ be the (unknown) conditional distribution of labels for a problem to be solved. For a parameterized conditional distribution $p(y|x, w)$ with parameters $w \in \mathbb{R}^D$, probabilistic parameter learning is the task of finding a point estimate of the parameter w^* that makes $p(y|x, w^*)$ closest to $d(y|x)$.

- We will discuss probabilistic parameter learning in detail.

Probabilistic Learning

Problem (Probabilistic Parameter Learning)

Let $d(y|x)$ be the (unknown) conditional distribution of labels for a problem to be solved. For a parameterized conditional distribution $p(y|x, w)$ with parameters $w \in \mathbb{R}^D$, probabilistic parameter learning is the task of finding a point estimate of the parameter w^* that makes $p(y|x, w^*)$ closest to $d(y|x)$.

- We will discuss probabilistic parameter learning in detail.

Loss-Minimizing Parameter Learning

Problem (Loss-Minimizing Parameter Learning)

Let $d(x, y)$ be the unknown distribution of data in labels, and let $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Loss minimizing parameter learning is the task of finding a parameter value w^* such that the expected prediction risk

$$\mathbb{E}_{(x,y)\sim d(x,y)}[\Delta(y, f_p(x))]$$

is as small as possible, where $f_p(x) = \operatorname{argmax}_{y\in\mathcal{Y}} p(y|x, w^*)$.

- Requires loss function at training time
- Directly learns a prediction function $f_p(x)$

Loss-Minimizing Parameter Learning

Problem (Loss-Minimizing Parameter Learning)

Let $d(x, y)$ be the unknown distribution of data in labels, and let $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Loss minimizing parameter learning is the task of finding a parameter value w^* such that the expected prediction risk

$$\mathbb{E}_{(x,y)\sim d(x,y)}[\Delta(y, f_p(x))]$$

is as small as possible, where $f_p(x) = \operatorname{argmax}_{y\in\mathcal{Y}} p(y|x, w^*)$.

- Requires loss function at training time
- Directly learns a prediction function $f_p(x)$