Lecture 9: Semantic Segmentation and CRFs

- Semantic Segmentation is an important application of Deep Networks, which assigns a class label to all pixels in the image. This was one of the first uses of Deep Networks after Object Classification. It was much more effective than pervious methods (e.g., statistical edge detection lecture). It contained many filters, able to capture the varieties of textures of the objects and background classes.
- This was applied to classification on the CoCo and Pascal-Segmentation datasets. Here the classes are either object labels (e.g., Car, Cat) and a single background class. It was also applied to Pascal-Context where the classes include objects but also a variety of background classes (e.g., sky, water, road). It can be extended to segmenting objects into their constituent parts, e.g., on the Pascal-Parts dataset.It can also be applied to segmenting organs and tumors in medical images.
- Semantic segmentation required datasets with per-pixel annotation. But it was realized that this could be supplemented with data that was only partially annotated. For example, where a box was specified to be a "foreground class", e.g., "cat", although it also contained non-cat background pixels (perhaps fifty percent pixels was "foreground", e.g., "cat", and the rest was background). This could be formulated by an EM algorithm in the natural way (e.g., each pixel was assigned a hidden variable which specified whether it was foreground or background). Recall that Deep Nets are formulated as performing maximum likelihood estimating, so we are simply extending this to include hidden variables

MRFs (1)

- Deep Networks for Semantic Segmentation (e.g., DeepLab) give estimates for the class labels of each pixel. But this can ignore the spatial context, neighboring pixels are likely to have the same labels. This spatial context (or temporal context) can be modeled by Markov Random Fields (MRFs). Here we will a brief introduction.
- We specify a posterior probability distribution $P(\mathbf{x}|\mathbf{z})$ defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the set of nodes $\mathcal{V}$ is the set of image pixels $\mathcal{D}$ and the edges $\mathcal{E}$ are between neighbouring pixels.
- The $\mathbf{x} = \{x_i : i \in \mathcal{V}\}$ are random variables specified at each node of the graph. $P(\mathbf{x}|\mathbf{z})$ is a Gibbs distribution specified by an energy function $E(\mathbf{x}, \mathbf{z})$.
- The energy function contains unary potentials $U(\mathbf{x}, \mathbf{z}) = \sum_{i \in \mathcal{V}} \phi(x_i, \mathbf{z})$ and pairwise potentials $V(\mathbf{x}, \mathbf{x}) = \sum_{ij \in \mathcal{E}} \psi_{ij}(x_i, x_j)$. The unary potentials $\phi(x_i, \mathbf{z})$ depend only on the label/disparity at node/pixel i and the dependence on the input $\mathbf{z}$ will depend on the application:

MRFs (2)

- In summary, the model is specified by a distribution $P(\mathbf{x}|\mathbf{z})$ defined over discrete-valued random variables $\mathbf{x} = \{x_i : i \in \mathcal{V}\}$ defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: $P(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \exp\{-\sum_{i \in \mathcal{V}} \phi_i(x_i, \mathbf{z}) - \sum_{ij \in \mathcal{E}} \psi_{ij}(x_i, x_j)\}$.
- The goal will be to estimate properties of the distribution such as the MAP estimator and the marginals (which relate to each other, as discussed later).

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{z}), \text{ the MAP estimate,}$$
$$p_i(x_i) = \sum_{\mathbf{x}/i} P(\mathbf{x}|\mathbf{z}), \ \forall i \in \mathcal{V} \text{ the marginals.} \quad (2)$$

MRFs: Mean Field Theory (1)

- We know describe the mean field algorithm for approximate inference.
- We define the mean field free energy $\mathcal{F}_{\mathrm{MFT}}(\mathbf{b})$ by:

$$\mathcal{F}_{\mathrm{MFT}}(\mathbf{b}) = \sum_{ij\in\mathcal{E}} \sum_{x_i,x_j} b_i(x_i) b_j(x_j) \psi_{ij}(x_i, x_j)$$

$$+ \sum_{i\in\mathcal{V}} \sum_{x_i} b_i(x_i) \phi_i(x_i, \mathbf{z}) + \sum_{i\in\mathcal{V}} \sum_{x_i} b_i(x_i) \log b_i(x_i). \tag{3}$$

- The first two terms are the expectation of the energy $E(\mathbf{x}, \mathbf{z})$ with respect to the distribution $\mathbf{b}(\mathbf{x})$ and the third term is the negative entropy of $\mathbf{b}(\mathbf{x})$. If the labels can take only two values – i.e. $x_i \in \{0, 1\}$ – then the entropy can be written as $\sum_{i\in\mathcal{V}} \{b_i \log b_i + (1 - b_i) \log(1 - b_i)\}$ where $b_i = b_i(x_i = 1)$. If the labels take a set of values $l = 1, .., N$, then we can express the entropy as $\sum_{i\in\mathcal{V}} \sum_{l=1}^{M} b_{il} \log b_{il}$ where $b_{il} = b_i(x_i = l)$ and hence the $\{b_{il}\}$ satisfy the constraint $\sum_{l=1}^{M} b_{il} = 1, \ \forall i$.

MRFs: Mean Field Theory (2)

- Justification for the Mean Field Free Energy,
- Substituting $P(\mathbf{x}) = \frac{1}{Z}\exp\{-E(\mathbf{x})\}$ and $B(\mathbf{x}) = \prod_{i\in\mathcal{V}} b_i(x_i)$ into the Kullback-Leibler divergence $KL(B,P)$ gives:
  $KL(B,P) = \sum_{\mathbf{x}} B(\mathbf{x})E(\mathbf{x}) + \sum_{\mathbf{x}} B(\mathbf{x})\log B(\mathbf{x}) + \log Z = \mathcal{F}_{\mathrm{MFT}}(B) + \log Z.$
- Hence minimizing $\mathcal{F}_{\mathrm{MFT}}(B)$ with respect to B gives: (i) the best factorized approximation to $P(\mathbf{x})$, and (ii) a lower bound to the partition function $\log Z \geq \min_B \mathcal{F}_{\mathrm{MFT}}(B)$ which can be useful to assess model evidence

MRFs: Mean Field Theory (3)

- Steepest Descent – or Discrete Update (Add – where is the latex file for the Book Chapter?).
- The mean field free energies are functions of continuous variables (since discrete variables have been replaced by continuous probability distributions) which enables us to compute gradients of the free energy. This allows us to use steepest descent algorithms and its many variants. Suppose we take the MFT free energy from equation (3), restrict $x_i \in \{0, 1\}$, set $b_i = b_i(x_i = 1)$, then basic steepest descent can be written as:

$$\begin{aligned} \frac{db_i}{dt} &= -\frac{\partial \mathcal{F}_{\mathrm{MFT}}}{\partial b_i}, \\ &= 2\sum_j \sum_{x_j} \psi_{ij}(x_i, x_j) b_j + \phi_i(x_i) - \{b_i \log b_i + (1 - b_i)\log(1 - b_i)\}. \end{aligned} \tag{4}$$

- The MFT free energy decreases monotonically because $\frac{d\mathcal{F}_{\mathrm{MFT}}}{dt} = \sum_i \frac{\partial \mathcal{F}_{\mathrm{MFT}}}{\partial b_i}\frac{db_i}{dt} = -\sum_i \{\frac{\partial \mathcal{F}_{\mathrm{MFT}}}{\partial b_i}\}^2$ (note that the energy decreases very slowly for small gradients – because the square of a small number is very small). The negative entropy term $\{b_i \log b_i + (1 - b_i)\log(1 - b_i)\}$ is guaranteed to keep the values of b_i within the range $[0, 1]$ (since the gradient of the negative entropy equals $\log b_1/(1 - b_i)$ which becomes infinitely large as $b_i \mapsto 0$ and $b_i \mapsto 1$).

MRFs: Mean Field Theory (4)

- In practice, we use a discrete approximation of form $b_i^{t+1} = b_i^t - \Delta \frac{\partial \mathcal{F}_{\mathrm{MFT}}}{\partial b_i}$, where b_i^t is the state at time t, but this requires a good choice of Δ.
- A simple variant, which has often been used, is to multiply the free energy gradient $\frac{\partial \mathcal{F}_{\mathrm{MFT}}}{\partial b_i}$ by a positive function (which still ensures that the free energy decreases monotonically). A typical choice of function is $b_i(1 - b_i)$ which, interestingly, gives dynamics which are identical to models of artificial neural networks.
- There is a related class of *discrete iterative algorithms* which can be expressed in form $b^{t+1} = f(b^t)$ for some function $f(.)$. They have two advantages over steepest descent algorithms: (i) they are guaranteed to decrease the free energy monotonically (i.e. $\mathcal{F}_{MFT}(b^{t+1}) \leq \mathcal{F}_{MFT}(b^t)$), and (ii) they are non-local so that b^{t+1} may be distant from b^t which can enable to escape some of the local minima which can trap steepest descent. Algorithms of this type can be derived by using principles such as variational bounding and CCCP (see earlier lectures).