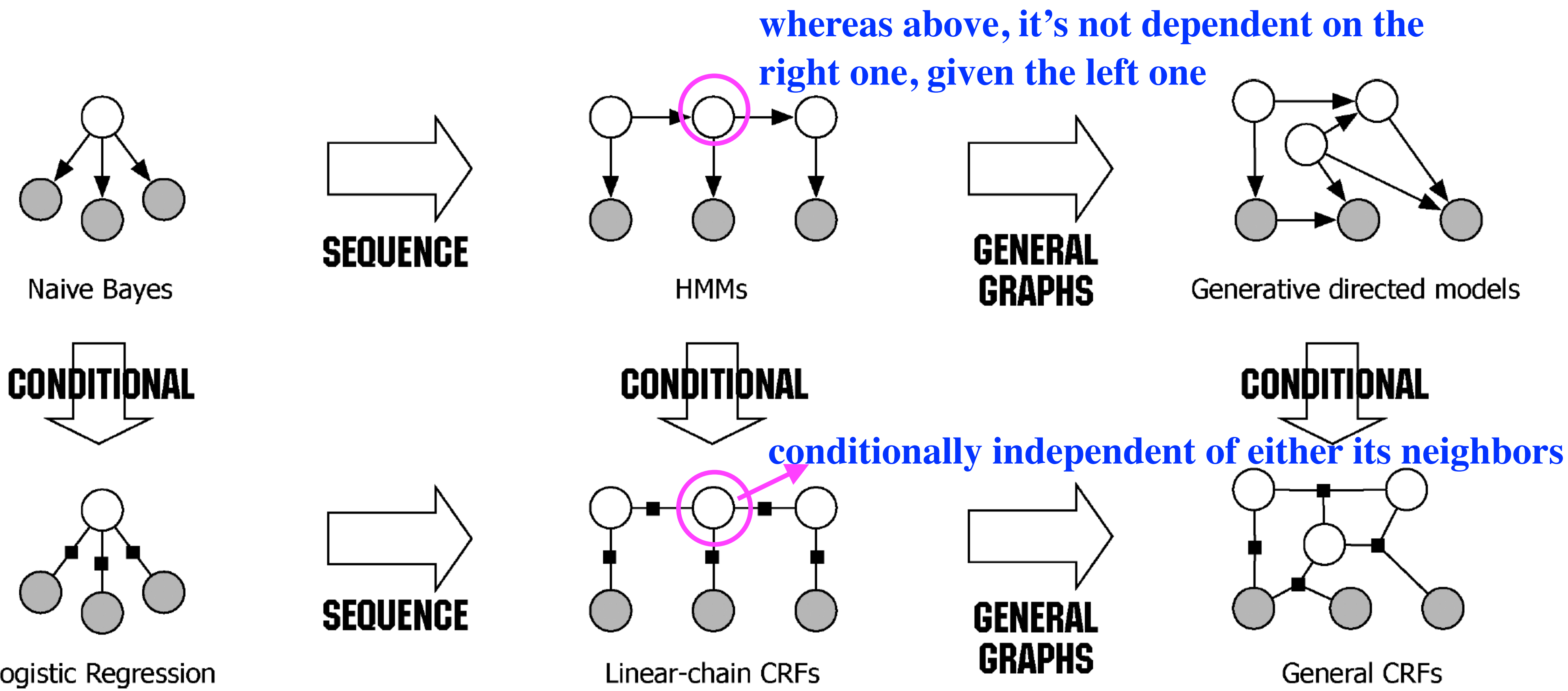
Conditional random fields

upper row — generative model: model the prior probabilities of things

we're not asking how likely is it in general that i'd get a certain label. we can only answer the question: given these inputs, what's the probability of that label. we can't even in that case, answer the question given a label, what are the inputs?

— — more complicated in the sense that we have a whole joint distribution

lower row — discriminative world: model the conditional probabilities



whereas above, it’s not dependent on the right one, given the left one

conditionally independent of either its neighbors

Fig. 2.4 Diagram of the relationship between naive Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

Charles Sutton. An Introduction to Conditional Random Fields. Foundations and Trends in Machine Learning, 4(4):267–373, 2012.

CRF vs. Bayesian network

- discriminative
    - in contrast with generative models, CRF models only the conditional distribution $p(y|x)$
- undirected
- potentially cyclic

linear-chain CRF vs. HMM

- each label may be correlated with all observations
- feature functions may be arbitrary (differentiable)

HMM, restated

$$
\begin{aligned}
p(\mathbf{y}, \mathbf{x}) &= \frac{1}{Z} \prod_{t=1}^{T} \exp \left\{ \sum_{i,j \in S} \theta_{ij} \mathbf{1}_{\{y_t = i\}} \mathbf{1}_{\{y_{t-1} = j\}} + \sum_{i \in S} \sum_{o \in O} \mu_{oi} \mathbf{1}_{\{y_t = i\}} \mathbf{1}_{\{x_t = o\}} \right\} \\
&= \frac{1}{Z} \prod_{t=1}^{T} \exp \left\{ \sum_{i,j \in S} \theta_{ij} f_{ij}(y_t, y_{t-1}, x_t) + \sum_{i \in S} \sum_{o \in O} \mu_{oi} f_{io}(y_t, y_{t-1}, x_t) \right\} \\
&= \frac{1}{Z} \prod_{t=1}^{T} \exp \left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right\}
\end{aligned}
$$

Linear-chain CRF

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp \left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}$$

Tasks

- can perform "decoding" - finding most likely state sequence via Viterbi
- can perform "inference" - finding best parameters θ_k via gradient descent

Application

feature functions can be:

- manually engineered features
    - e.g. "word ends in *-ing*"
- neural networks
    - e.g. the popular BiLSTM + CRF