Learning Graphical models: Basic Framework

- Generating a graphical model by hand, expert, etc. is not possible
  - Thousands and sometimes millions of variables (e.g., the Web domain)

- **Input:** A data set \( \mathcal{X} = \{x[1], \ldots, x[M]\} \) having \( M \) examples or samples. (Assumption: the \( M \) examples are independent and identically distributed (IID); generated from \( P^* \))

- **Output:** A graphical model \( \tilde{M} \) representing a distribution \( \tilde{P} \) such that
  - it is as close as possible to \( P^* \)
  - it solves the task/problem that you are interested in as accurately as possible
Evaluating Learning Performance

- Given candidate models, how do I evaluate which is better?
  - Non-trivial task. How do I define the **notion of best**?
- Various performance metrics depending upon your learning goal.
Performance metric 1: Relative Entropy or KL distance

\[ D(P^* \| \tilde{P}) = \sum_x P^*(x) \log(P^*(x)) - \sum_x P^*(x) \log(\tilde{P}(x)) \]

- We can not evaluate this directly (exponentially large). However, we can consider our data as samples and use the following Monte Carlo estimate:

\[ \hat{D}(P^* \| \tilde{P}) = \frac{1}{M} \sum_{i=1}^{M} \log(P^*(x^{(i)})) - \frac{1}{M} \left[ \sum_{i=1}^{M} \log(\tilde{P}(x^{(i)})) \right] \]

- The first term is a constant (no need to evaluate). The term in [...] is called the log-likelihood of the data.
Performance metric 1: Maximum likelihood learning (MLE)

\[
\hat{D}(\hat{P}^\ast || \hat{P}) = \frac{1}{M} \sum_{i=1}^{M} \log(P^\ast(x^{(i)})) - \frac{1}{M} \left[ \sum_{i=1}^{M} \log(\hat{P}(x^{(i)})) \right]
\]

- We should prefer models that have the maximum value for \(\sum_{i=1}^{M} \log(\hat{P}(x^{(i)}))\) (log-likelihood)
- Will likely minimize the error (i.e., improve accuracy)
- Since logarithm is monotonic, maximizing the log-likelihood is same as maximizing the likelihood:

\[
L(x^1, \ldots, x^M) = \exp \left[ \sum_{i=1}^{M} \log(\hat{P}(x^{(i)})) \right] = \prod_{i=1}^{M} \hat{P}(x^{(i)})
\]
Performance metric 2: Task directed learning

- You may be interested in a specific task
  - Classification task: Given a set of documents, find the topic of each document
    - Classification error: \# of mis-classified instances.
    - Hamming loss: When we are interested in multi-class labeling, we count the number of variables that are mis-classified
  - Query Variables \( Y \): You may be interested in querying only a subset of the variables. Let the other variables \( X \setminus Y \) be denoted by \( Z \).
    - Maximize conditional log likelihood:
    \[
    \sum_{y,z} P^*(y, z) \log(\tilde{P}(y|z))
    \]
Basic Machine learning Concepts: Review

- **Overfitting**: the learned model to the training set. Extreme example: The data is the model.
- **Generalization**: the data is a sample, there is vast amount of samples that you have never seen. Your model should generalize well to these “never-seen” samples.
- **Bias-Variance tradeoff**: Richer vs constrained models. Example: high treewidth vs low treewidth models
  - Can learn low treewidth models (Example: learning trees is easy). However, a tree may not represent all independencies of $P^*$ (not a minimal I-map).
  - Cannot learn high treewidth models (limited data). However, they may be closer to $P^*$. 

Regularization: Encode a soft constraint for simpler models in our objective function.

Note: Restricting our model class reduces over-fitting. This imposes a hard constraint. Regularization is a soft constraint.

Training versus Test-set: Hold out some data as test data.

k-fold cross validation: A special way of holding out data. Divide the data into k bins. Run your algorithm k times. Each time use the i-th bin as test data.
Basic Machine learning Concepts: Review

- **PAC bounds**: Formal analysis of how good your learning algorithm is: Let $\epsilon > 0$ be our approximation parameter, $\delta > 0$ be our confidence parameter and $M$ be the size of our data set $\mathcal{X}$. Then, for $M$ large enough, we have:

$$\Pr(D(P^* \| P_{\tilde{M},\mathcal{X}}) \leq \epsilon) \geq 1 - \delta$$

- For sufficiently large $M$, for most data sets of size $M$ sampled from $P^*$, the learning procedure will yield a close approximation to $P^*$.

- **Sample Complexity**: The size of $M$ that is required to guarantee this bound.
Learning task definition revisited

Given:
- Some prior knowledge or constraints on $\tilde{M}$
- A data set: $\mathcal{X} = \{\mathbf{x}[1], \ldots, \mathbf{x}[M]\}$; samples generated iid from $P^*$.

Output
- A graphical model $\tilde{M}$ which may include the structure (structure learning), the parameters (parameter learning) or both.
Data Observability

- **Fully observed**: Complete data so that each of our training instances is an assignment of values to all variables.
- **Partially observed**: There exists training instances $t$ such that one or more variables in $t$ are not observed (missing values).
- **Hidden variables**: The data contains hidden variables whose value is never observed.