Variational Bayesian Learning

Tapani Raiko
April 17, 2008

Machine learning: Advanced probabilistic methods
Motivation

- The main issue in probabilistic machine learning models is to find the posterior distribution over the model parameters and latent variables.
- EM uses a point estimate for parameters which may be prone to over-fitting. Also, the E-step may not be solvable for some models.
- Sampling is prohibitively slow for large latent variable models.
- Variational Bayesian (VB) learning is a good compromise.
Overfitting

• An overfitted model explains the current data but does not generalize well to new data

• 6th order polynomial is fitted to 10 points by maximum likelihood and sampling
Posterior mass matters

- You want to make predictions about new data $Y$ based on existing data $X$

- This is solved by fitting a model to the data and then predicting based on that

$$p(Y \mid X) = \int p(Y \mid X, Z, \theta)p(Z, \theta \mid X)dZd\theta$$

- Note how you need to integrate over the posterior $p(Z, \theta \mid X)$

- If you need to select a single solution $Z, \theta$, it should represent the posterior mass well
Why early stopping might help
Example: Probabilistic Principal Component Analysis (PCA)

\[ x_j = A s_j + \epsilon_j . \]
\[ p(s_j) = \mathcal{N} (s_j; 0, I) , \quad p(\epsilon_j) = \mathcal{N} (\epsilon_j; 0, \nu I) \]

- Continuous-valued data vectors \( x \) are modelled as a linear mixture of source vectors \( s \) and noise
- Traditional PCA is the case where the noise goes to zero
Recap: EM-algorithm

- EM-algorithm solves latent variable models by alternating between two steps:
  - E-step updates the distribution over the latent variables $Z$
  - M-step updates the estimate of parameters $\theta$

E-step: $Q(Z) \leftarrow P(Z \mid X, \theta)$

M-step: $\theta \leftarrow \arg \max_\theta E_{Q(Z)} \{ \ln P(X, Z \mid \theta) \}$
EM for PPCA
(don’t learn the formulas by heart)

- The source posterior is a Gaussian:
  \[ p(S|X, A, v) = \prod_{j=1}^{n} \mathcal{N}(s_j; \bar{s}_j, \Sigma_s) \]

- E-step:
  \[ \bar{S} = \Psi^{-1} A^T X, \quad \Sigma_s = v \Psi^{-1}, \quad \Psi = A^T A + v I. \]

- M-step:
  \[ A = X S^T (n \Sigma_s + S S^T)^{-1} \]
  \[ v = \frac{1}{nd} \sum_{i=1}^{d} \sum_{j=1}^{n} (x_{ij} - a_i^T \bar{s}_j)^2 + \frac{1}{d} \text{tr}(A \Sigma_s A^T). \]
The model equation $X=AS$ is symmetric with respect to $A$ and $S$

Why are $A$ and $S$ treated so differently?

Would it be possible to model the posterior of both $A$ and $S$ with a Gaussian?
VB-EM algorithm

- The VB-EM algorithm alternates between updates for the latent variables and parameters

- Steps are symmetric and they resemble the E-step of the EM algorithm

- **VB-E step:**
  
  \[
  q(Z) \leftarrow \text{argmin}_{q(Z)} E_{q(\theta)} \{ KL (q(Z) \parallel p(Z | X, \theta)) \}
  \]

- **VB-M step:**
  
  \[
  q(\theta) \leftarrow \text{argmin}_{q(\theta)} E_{q(Z)} \{ KL (q(\theta) \parallel p(\theta | X, Z)) \}
  \]
Variational Bayes (key slide!)

- VB works by fitting a distribution $q$ over the unknown variables to the true posterior by minimizing the KL divergence:

$$\text{KL} (q(Z, \theta) \parallel p(Z, \theta \mid X)) = E_{q(Z,\theta)} \left\{ \ln \frac{q(Z, \theta)}{p(Z, \theta \mid X)} \right\}$$

- The form of $q$ can be chosen such that the expectations are tractable

- For instance, $q(Z, \theta) = q(Z)q(\theta)$ is assumed almost always, allowing the VB-EM algorithm
Example 1

- **model**
  \[ p(z) = \mathcal{N}(z; xy, 0.02) \]

- **prior**
  \[ p(x) = \mathcal{N}(x; 0, 1), \]
  \[ p(y) = \mathcal{N}(y; 0, 1). \]

- **data**
  \[ z = 1 \]
Example 2

- model
  \[ p(z) = \mathcal{N}(z; y, \exp(-x)) \]
- prior
  \[ p(x) = \mathcal{N}(x; -1, 5) \]
  \[ p(y) = \mathcal{N}(y; 0, 5) \]
- data
  \[ z = 2 \]
VB-EM for PCA
(don’t learn the formulas by heart)

\[ q(A, S) = \prod_{i=1}^{d} N(a_i; \overline{a}_i, \Sigma_a) \prod_{j=1}^{n} N(s_j; \overline{s}_j, \Sigma_s). \]

\[ \overline{S} = \Psi^{-1} \overline{A}^T X, \quad \Sigma_s = \nu \Psi^{-1} \]

\[ \Psi = \overline{A}^T \overline{A} + d \Sigma_a + \nu I. \]

\[ \overline{A} = \Phi^{-1} \overline{S} X, \quad \Sigma_a = \nu \Phi^{-1} \]

\[ \Phi = \overline{S} \overline{S}^T + n \Sigma_s + \nu \text{diag}(w_k^{-1}) \]

\[ \nu = \frac{1}{nd} \sum_{i=1}^{d} \sum_{j=1}^{n} (x_{ij} - \overline{a}_i \overline{s}_j)^2 + \frac{1}{d} \text{tr}(\overline{A} \Sigma_s \overline{A}^T) \frac{1}{n} \text{tr}(\overline{S}^T \Sigma_a \overline{S}) + \frac{1}{nd} \text{tr}(\Sigma_s \Sigma_a). \]
Compare to EM

• The source posterior is a Gaussian:

\[ p(S|X, A, v) = \prod_{j=1}^{n} \mathcal{N}(s_j; \bar{s}_j, \Sigma_s) \]

• E-step:

\[ \bar{S} = \Psi^{-1} A^T X, \quad \Sigma_s = v \Psi^{-1}, \quad \Psi = A^T A + v I. \]

• M-step:

\[ A = X S^T (n \Sigma_s + SS^T)^{-1} \]

\[ v = \frac{1}{nd} \sum_{i=1}^{d} \sum_{j=1}^{n} (x_{i,j} - a_i^T \bar{s}_j)^2 + \frac{1}{d} \text{tr}(A \Sigma_s A^T). \]
Model selection

• The cost function that is minimized in practice is also includes a part for model evidence $p(X|M)$

$$C_{VB} = E_q \left\{ \ln \frac{q(Z, \theta)}{p(X, Z, \theta | M_i)} \right\}$$

$$= KL (q(Z, \theta) \parallel p(Z, \theta | X, M_i)) - \ln p(X | M_i)$$

$$\geq - \ln p(X | M_i)$$

• By minimizing the cost, we get a lower bound for the model evidence

• We can thus compare different models $M$
Learning algorithms

• q can be parameterized for instance by posterior means and covariances

• Those variational parameters can then be updated by any means to minimize the cost

\[ C_{VB} = E_q \left\{ \ln \frac{q(Z, \theta)}{p(X, Z, \theta | M_i)} \right\} \]

• This is useful if the VB-EM updates are intractable

• Gradient based methods can be faster, too
Discrete models

• Consider VB learning of Bayesian networks
• Instead of a single set of parameters (conditional probability tables), we would have distribution $q(\theta)$ over the parameters
• The certainty of CPTs would be estimated
• The VB cost function could be used to select the best model structure (it penalizes complex models automatically)
• By restricting the form of $q(Z)$, the inference (E-step) can be made faster.
Pros and cons of VB

- Robust against overfitting
- Fast (compared to sampling)
- Applicable to a large family of models
- Intensive formulae (lots of integrals)
- Prone to bad but locally optimal solutions (lot of work with arranging good initializations and other tricks to avoid them)
Software packages for VB on Bayesian networks (1/2)

- VIBES by Winn and Bishop
  - discrete and continuous values
  - posterior approximation is factorized such that disjoint groups of variables are independent but dependencies within the group are modelled
  - variational message passing algorithm
Software packages for VB on Bayesian networks (2/2)

• Bayes Block by Valpola et al.
  • concentrates on continuous values
  • fully factorial posterior approximation
  • includes nonlinearities
  • allows for variance modelling
  • message passing with line searches for speed-up