Stochastic Discriminative EM (sdEM)

Discriminative Learning in the Natural Exponential Family

Andrés R. Masegosa

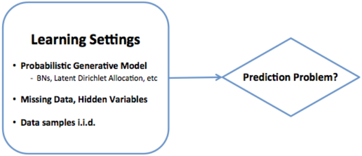
August 5, 2014

The BIG picture

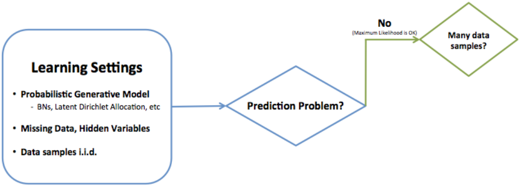
Learning Settings

- **Probabilistic Generative Model**
  - BNs, Latent Dirichlet Allocation, etc
- **Missing Data, Hidden Variables**
- **Data samples i.i.d.**

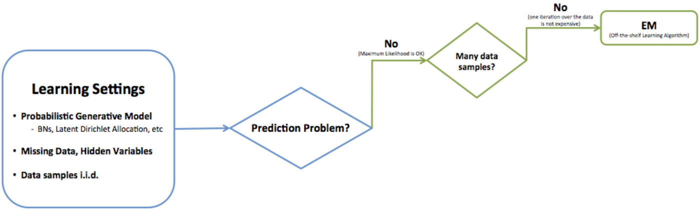
The BIG picture



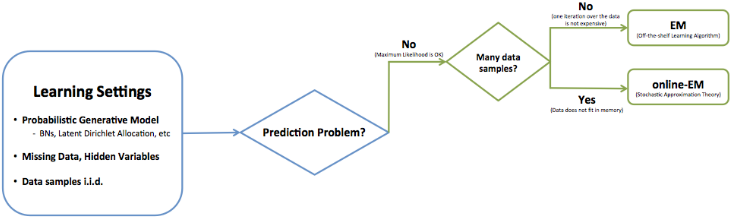
The BIG picture



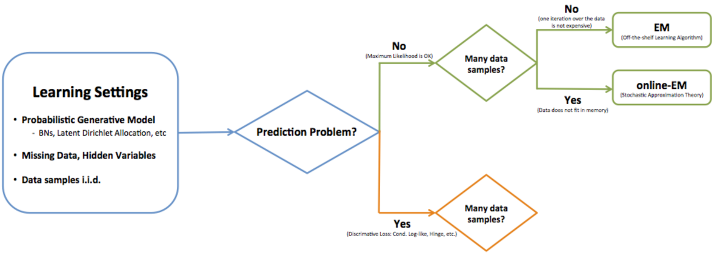
The BIG picture



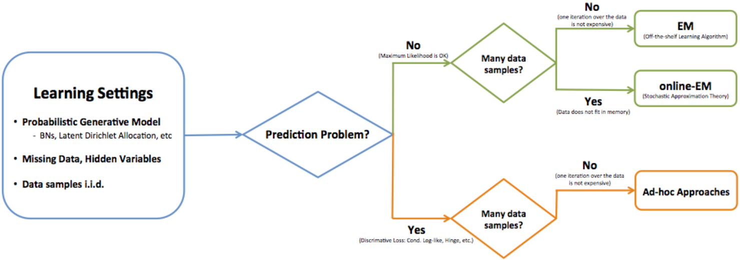
The BIG picture



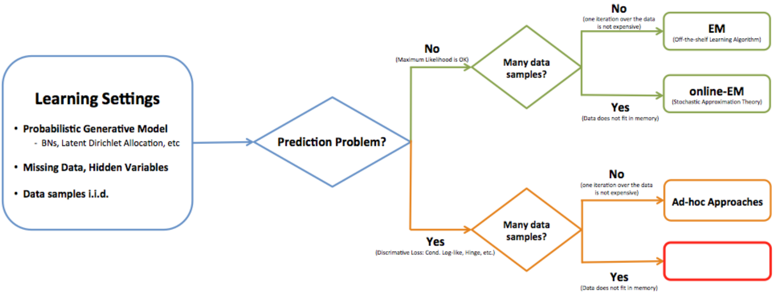
The BIG picture



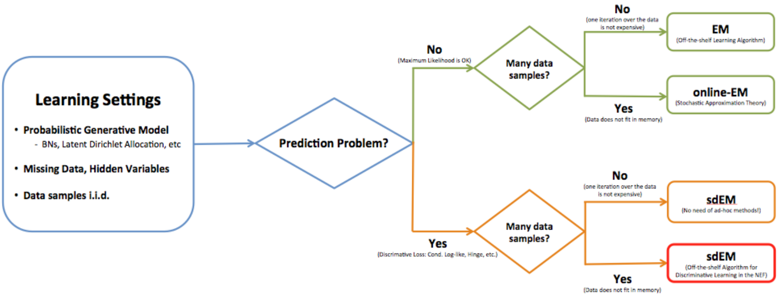
The BIG picture



The BIG picture



The BIG picture



Definitions and Notation

- **Prediction Problem**:
  - Y variable to be predicted (discrete, continuous or vector-value).
  - X the predictive variables.

Definitions and Notation

- **Prediction Problem**:
  - Y variable to be predicted (discrete, continuous or vector-value).
  - X the predictive variables.
- $p(y, x|\theta)$ in **the natural exponential family**.

Definitions and Notation

- **Prediction Problem**:
  - Y variable to be predicted (discrete, continuous or vector-value).
  - X the predictive variables.
- $p(y, x|\theta)$ in **the natural exponential family**.
- **Generative Learning (maximum likelihood in a generative model)**:
  - Given data $D = \{(y_1, x_1), ..., (y_N, x_N)\}$ solve the problem

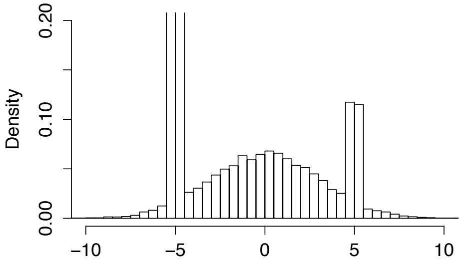
$$arg \max_{\theta} \sum_{(y_i, x_i) \in D} \ln p(y_i, x_i|\theta)$$

Definitions and Notation

- **Prediction Problem**:
  - Y variable to be predicted (discrete, continuous or vector-value).
  - X the predictive variables.
- $p(y, x|\theta)$ in **the natural exponential family**.
- **Generative Learning (maximum likelihood in a generative model)**:
  - Given data $D = \{(y_1, x_1), ..., (y_N, x_N)\}$ solve the problem

$$arg \min_{\theta} \sum_{(y_i, x_i) \in D} - \ln p(y_i, x_i|\theta)$$

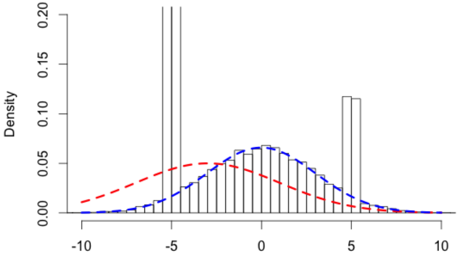
Maximum Likelihood Estimation



Distribution of the data $\pi(x, y) = \pi(x|y)\pi(y)$:

- Two classes with equal prior: $\pi(y = -1) = \pi(y = 1) = 0.5$
- Negative class is Gaussian distributed: $\pi(x|y = -1) \sim N(0, 3)$
- Positive class is a mixture of Gaussians:
  $\pi(x|y = 1) \sim 0.8 \cdot N(-5, 0.1) + 0.2 \cdot N(5, 0.1)$

Maximum Likelihood Estimation



Generative Learning or Maximum Likelihood:

- The model to be fitted is $p(y, x)$ assumes $p(x|y)$ is univariate Gaussian.
- Prediction Accuracy around 78%

A new look at maximum likelihood estimation

ID	X
1	0
2	0
3	1
4	0
5	1

1. Counting...

- $n_{i+1}^{(0)} = n_i^{(0)} + I[x_i == 0]$
- $n_{i+1}^{(1)} = n_i^{(1)} + I[x_i == 1]$

and finally normalize $\bar{n}_N^{(0)} = n_N^{(0)}/N$.

A new look at maximum likelihood estimation

ID	X	$n_i^{(0)}$	$n_i^{(1)}$
1	0	1	0
2	0	2	0
3	1	2	1
4	0	3	1
5	1	3	2
		3/5	2/5

1. Counting...

- $n_{i+1}^{(0)} = n_i^{(0)} + I[x_i == 0]$
- $n_{i+1}^{(1)} = n_i^{(1)} + I[x_i == 1]$

and finally normalize $\bar{n}_N^{(0)} = n_N^{(0)} / N$.

A new look at maximum likelihood estimation

ID	X	$n_i^{(0)}$	$n_i^{(1)}$
1	0	1	0
2	0	2	0
3	1	2	1
4	0	3	1
5	1	3	2
		3/5	2/5

1. Counting...

- $n_{i+1}^{(0)} = n_i^{(0)} + I[x_i == 0]$
- $n_{i+1}^{(1)} = n_i^{(1)} + I[x_i == 1]$

and finally normalize $\bar{n}_N^{(0)} = n_N^{(0)}/N$.

2. Compute parameters from countings

- $\theta^{(0)} = \bar{n}_N^{(0)}/(\bar{n}_N^{(0)} + \bar{n}_N^{(1)})$
- $\theta^{(1)} = \bar{n}_N^{(1)}/(\bar{n}_N^{(0)} + \bar{n}_N^{(1)})$

A new look at maximum likelihood estimation

ID	X	$n_i^{(0)}$	$n_i^{(1)}$
1	0	1/1	0/1
2	0	2/2	0/2
3	1	2/3	1/3
4	0	2/4	2/4
5	1	3/5	2/5
		3/5	2/5

1. Normalized counting:

- $\bar{n}_{t+1}^{(0)} = (1-\rho_t)\bar{n}_t^{(0)} + \rho_t I[x_i == 0]$
- $\bar{n}_{t+1}^{(1)} = (1-\rho_t)\bar{n}_t^{(1)} + \rho_t I[x_i == 1]$

where $\rho_t = \frac{1}{t}$.

2. Compute parameters from countings

- $\theta^{(0)} = \bar{n}_N^{(0)} / (\bar{n}_N^{(0)} + \bar{n}_N^{(1)})$
- $\theta^{(1)} = \bar{n}_N^{(1)} / (\bar{n}_N^{(0)} + \bar{n}_N^{(1)})$

A new look at maximum likelihood estimation

- $\bar{n}^{(0)}$ **and** $\bar{n}^{(1)}$ **can also parameterize** $P(X|\bar{n}^{(0)}, \bar{n}^{(1)})$
  - 1-to-1 relation with θ parameters.
  - They are called the **expectation parameters**.

A new look at maximum likelihood estimation

- $\bar{n}^{(0)}$ **and** $\bar{n}^{(1)}$ **can also parameterize** $P(X|\bar{n}^{(0)}, \bar{n}^{(1)})$
    - 1-to-1 relation with θ parameters.
    - They are called the **expectation parameters**.

- Normalized counting is an **iterative updating of the expectation parameters**:

$$\bar{n}^{(0)}_{t+1} = (1 - \rho_t)\bar{n}^{(0)}_t + \rho_t I[x_i == 0]$$

A new look at maximum likelihood estimation

- $\bar{n}^{(0)}$ **and** $\bar{n}^{(1)}$ **can also parameterize** $P(X|\bar{n}^{(0)}, \bar{n}^{(1)})$
  - 1-to-1 relation with θ parameters.
  - They are called the **expectation parameters**.

- Normalized counting is an **iterative updating of the expectation parameters**:

$$\bar{n}^{(0)}_{t+1} = (1 - \rho_t)\bar{n}^{(0)}_t + \rho_t I[x_i == 0]$$

- **Compact notation**:

$$\bar{n}_{t+1} = (1 - \rho_t)\bar{n}_t + \rho_t s(x_t)$$

where $s(x) = (I[x == 0], I[x == 1])$ is the **sufficient statistics function**.

A new look at maximum likelihood estimation

- **After some maths....**:

$$\bar{n}_{t+1} \quad = \quad (1-\rho_t)\bar{n}_t + \rho_t s(x_t)$$

A new look at maximum likelihood estimation

- **After some maths....**:

$$\begin{aligned}\bar{n}_{t+1} &= (1-\rho_t)\bar{n}_t + \rho_t s(x_t) \\ &= \bar{n}_t + \rho_t \left(s(x_t) - \bar{n}_t\right)\end{aligned}$$

A new look at maximum likelihood estimation

- **After some maths....**:

$$
\begin{aligned}
\bar{n}_{t+1} &= (1-\rho_t)\bar{n}_t + \rho_t s(x_t) \\
&= \bar{n}_t + \rho_t \left(s(x_t) - \bar{n}_t \right) \\
&= \bar{n}_t + \rho_t \frac{\tilde{\partial} \ln p(x_t|\bar{n}_t)}{\tilde{\partial}\bar{n}}
\end{aligned}
$$

where $\tilde{\partial}$ denotes the *natural* gradient (Riemanian gemotry).

A new look at maximum likelihood estimation

- **After some maths....**:

$$\begin{aligned}\bar{n}_{t+1} &= (1-\rho_t)\bar{n}_t + \rho_t s(x_t)\\ &= \bar{n}_t + \rho_t\left(s(x_t) - \bar{n}_t\right)\\ &= \bar{n}_t + \rho_t \frac{\tilde{\partial} \ln p(x_t|\bar{n}_t)}{\tilde{\partial}\bar{n}}\end{aligned}$$

  where $\tilde{\partial}$ denotes the *natural* gradient (Riemanian gemotry).

- ....is equivalent to a **stochastic gradient ascent** method:
  - $\frac{\tilde{\partial} \ln p(x_t|\bar{n}_t)}{\tilde{\partial}\bar{n}}$ is a noisy estimate of the gradient of this function

$$f_D(\bar{n}) = \sum_{x_t \in D} \ln P(x_t|\bar{n})$$

A new look at maximum likelihood estimation

- **After some maths....**:

$$\begin{aligned}\bar{n}_{t+1} &= (1-\rho_t)\bar{n}_t + \rho_t s(x_t)\\ &= \bar{n}_t + \rho_t\left(s(x_t) - \bar{n}_t\right)\\ &= \bar{n}_t + \rho_t \frac{\tilde{\partial} \ln p(x_t|\bar{n}_t)}{\tilde{\partial}\bar{n}}\end{aligned}$$

  where $\tilde{\partial}$ denotes the *natural* gradient (Riemanian gemotry).

- ....is equivalent to a **stochastic gradient ascent** method:
  - $\frac{\tilde{\partial} \ln p(x_t|\bar{n}_t)}{\tilde{\partial}\bar{n}}$ is a noisy estimate of the gradient of this function

$$f_D(\bar{n}) = \sum_{x_t \in D} \ln P(x_t|\bar{n})$$

  - **Stochastic approximation theory** guarantees the convergence of the above iteration if

$$\sum_t \rho_t = \infty \quad \sum_t \rho_t^2 < \infty$$

Discriminative learning

- **The above algorithm also works for other loss functions**:

$$\bar{n}_{t+1} \quad = \quad \bar{n}_t - \rho_t \frac{\tilde{\partial}\ell(y_t, x_t|\bar{n}_t)}{\tilde{\partial}\bar{n}}$$

  - The convergence is guarantee by **stochastic approximation theory**.

Discriminative learning

- The **negative conditional log-likelihood**,

$$\ell(y_t, x_t|\bar{n}_t) = -\ln p(y_t|x_t, \bar{n}_t) = -\ln p(y_t, x_t|\bar{n}_t) + \ln p(x_t|\bar{n}_t)$$

Discriminative learning

- The **negative conditional log-likelihood**,

$$\ell(y_t, x_t|\bar{n}_t) = -\ln p(y_t|x_t, \bar{n}_t) = -\ln p(y_t, x_t|\bar{n}_t) + \ln p(x_t|\bar{n}_t)$$

- The **updating equation**:

$$\bar{n}_{t+1} \quad = \quad \bar{n}_t + \rho_t\,(s(y_t, x_t) - E_y[s(y, x_t)|\bar{n}_t])$$

Discriminative learning

- The **negative conditional log-likelihood**,

$$\ell(y_t, x_t|\bar{n}_t) = -\ln p(y_t|x_t, \bar{n}_t) = -\ln p(y_t, x_t|\bar{n}_t) + \ln p(x_t|\bar{n}_t)$$

- The **updating equation**:

$$\bar{n}_{t+1} \quad = \quad \bar{n}_t + \rho_t \left(s(y_t, x_t) - E_y[s(y, x_t)|\bar{n}_t]\right)$$

  - For a naive Bayes classifier, the iteration equations are simply expressed:

$$\bar{n}^{(0)}_{t+1} = \bar{n}^{(0)}_t + \rho_t(1 - p(y = 0|x_t)) \quad \text{if } y_t == 0.$$

$$\bar{n}^{(1)}_{t+1} = \bar{n}^{(1)}_t - \rho_t p(y = 1|x_t) \quad \text{if } y_t == 0.$$

Discriminative learning

- The **Hinge** or **max-margin** loss,

$$\ell_{hinge}(y_t, x_t, \theta) = \max(0, 1 - \ln \frac{p(y_t, x_t|\theta)}{p(\bar{y}_t, x_t|\theta)}) \tag{1}$$

where $\bar{y}_t$ denotes here too the most offending incorrect answer, $\bar{y}_t = arg \max_{y \neq y_t} p(y, x_t|\theta)$.

Discriminative learning

- The **Hinge** or **max-margin** loss,

$$\ell_{hinge}(y_t, x_t, \theta) = \max(0, 1 - \ln \frac{p(y_t, x_t|\theta)}{p(\bar{y}_t, x_t|\theta)}) \quad (1)$$

where $\bar{y}_t$ denotes here too the most offending incorrect answer, $\bar{y}_t = arg\max_{y \neq y_t} p(y, x_t|\theta)$.

- The **updating equation**:

$$\bar{n}_{t+1} = \bar{n}_t + \rho_t \left(s(y_t, x_t) - s(\bar{y}_t, x_t)\right) \quad \textit{if} \ \ln \frac{p(y_t, x_t|\theta)}{p(\bar{y}_t, x_t|\theta)} < 1$$

Discriminative learning

- The **Hinge** or **max-margin** loss,

$$\ell_{hinge}(y_t, x_t, \theta) = \max(0, 1 - \ln \frac{p(y_t, x_t|\theta)}{p(\bar{y}_t, x_t|\theta)}) \quad (1)$$

  where $\bar{y}_t$ denotes here too the most offending incorrect answer, $\bar{y}_t = arg\max_{y \neq y_t} p(y, x_t|\theta)$.

- The **updating equation**:

$$\bar{n}_{t+1} \quad = \quad \bar{n}_t + \rho_t\left(s(y_t, x_t) - s(\bar{y}_t, x_t)\right) \quad \text{if} \;\ln \frac{p(y_t, x_t|\theta)}{p(\bar{y}_t, x_t|\theta)} < 1$$

  - For a naive Bayes classifier, the iteration equations are simply expressed:

$$\bar{n}^{(0)}_{t+1} = \bar{n}^{(0)}_t + \rho_t \cdot 1 \quad \text{if } y_t == 0 \text{ and } \ln \tfrac{p(y_t, x_t|\theta)}{p(\bar{y}_t, x_t|\theta)} < 1$$

$$\bar{n}^{(1)}_{t+1} = \bar{n}^{(1)}_t - \rho_t \cdot 1 \quad \text{if } y_t == 0 \text{ and } \ln \tfrac{p(y_t, x_t|\theta)}{p(\bar{y}_t, x_t|\theta)} < 1$$

Discriminative learning with hidden variables

- The **p(y,z,x) is in the natural exponential family** and $s(y, z, x)$ the suff. stats.

Discriminative learning with hidden variables

- The **p(y,z,x) is in the natural exponential family** and $s(y, z, x)$ the suff. stats.

- The **negative conditional log-likelihood**,

$$\bar{n}_{t+1} = \bar{n}_t + \rho_t \left(E_z[s(y_t, z, x_t)|\bar{n}_t] - E_{yz}[s(y, z, x_t)|\bar{n}_t] \right)$$

Discriminative learning with hidden variables

- The **p(y,z,x) is in the natural exponential family** and $s(y,z,x)$ the suff. stats.

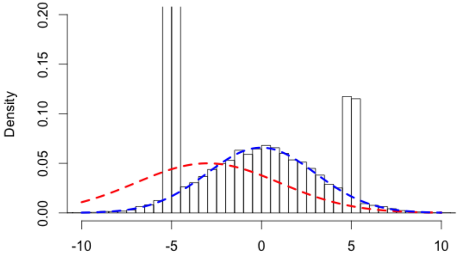
- The **negative conditional log-likelihood**,

$$\bar{n}_{t+1} = \bar{n}_t + \rho_t \left(E_z[s(y_t,z,x_t)|\bar{n}_t] - E_{yz}[s(y,z,x_t)|\bar{n}_t] \right)$$

- The **Hinge loss**:

$$\bar{n}_{t+1} = \bar{n}_t + \rho_t \left(E_z[s(y_t,z,x_t)|\bar{n}_t] - E_z[s(\bar{y}_t,z,x_t)|\bar{n}_t] \right)$$

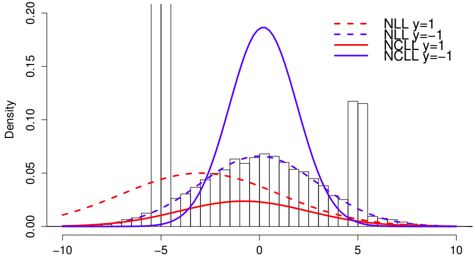
What is discriminative learning?



Generative Learning or Maximum Likelihood:

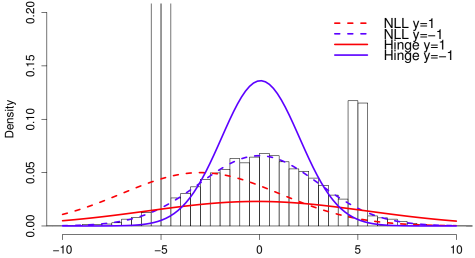
- The model to be fitted is $p(y, x)$ assumes $p(x|y)$ is univariate Gaussian.
- Prediction Accuracy around 78%

What is discriminative learning?



Discriminative Learning with the NCLL loss: 90.4% of accuracy

What is discriminative learning?



Discriminative Learning with the Hinge Loss: 90.6% of accuracy

What's the relationship with EM?

Algorithm 1 Standard EM

Choose some θ_0;
$t = 0$;
repeat
  $n_0 = 0$
  for $i = 1, \ldots, N$ **do**
    E-Step: $n_{i+1} = n_i + (E_z[s(y_i, z, x_i)|\theta_t])$
  end for
  $\bar{n}_t = n_N / N$
  M-Step: $\theta_{t+1} = \theta(\bar{n}_t)$;
  $t = t + 1$;
until convergence
return $\theta(\bar{n}_t)$;

What's the relationship with EM?

Algorithm 2 Standard EM

Choose some θ_0;
$t = 0$;
repeat
  $\bar{n}_0 = 0$
  for $i = 1, \ldots, N$ **do**
    E-Step: $\bar{n}_{i+1} = (1 - \frac{1}{i})\bar{n}_i + \frac{1}{i} \cdot (E_z[s(y_i, z, x_i)|\theta_t])$
  end for
  M-Step: $\theta_{t+1} = \theta(\bar{n}_N)$;
  $t = t + 1$;
until convergence
return $\theta(\bar{n}_t)$;

What's the relationship with EM?

Algorithm 3 Online EM

Require: D is randomly shuffled.

Choose some θ_0;
$t = 0$;
$\bar{n}_0 = 0$
repeat
  for $i = 1, \ldots, N$ **do**
    E-Step: $\bar{n}_{t+1} = (1 - \rho_t)\bar{n}_t + \rho_t \cdot (E_z[s(y_i, z, x_i)|\theta_t])$
    M-Step: $\theta_{t+1} = \theta(\bar{n}_t)$;
    $t = t + 1$;
  end for
until convergence
return $\theta(\bar{n}_t)$;

What's the relationship with EM?

Algorithm 4 sdEM with NCLL loss

Require: D is randomly shuffled.

Choose some θ_0;
$t = 0$;
$\bar{n}_0 = 0$
repeat
  for $i = 1, \ldots, N$ **do**
    E-Step: $\bar{n}_{t+1} = \bar{n}_t + \rho_t \cdot (E_z[s(y_i, z, x_i)|\theta_t] - E_{yz}[s(y_i, z, x_i)|\theta_t])$
    M-Step: $\theta_{t+1} = \theta(\bar{n}_t)$;
    $t = t + 1$;
  end for
until convergence
return $\theta(\bar{n}_t)$;

Some more details about sdEM

- Employment of a **conjugate prior** $p(\theta|\alpha)$

$$\textit{arg} \min_{\theta} \sum_{(y_i, x_i) \in D} \ell(y_i, x_i, \theta) + \ln p(\theta|\alpha)$$

  - Guarantees convergence: $\ln p(\theta|\alpha)$ is a log-barrier function.

Some more details about sdEM

- Employment of a **conjugate prior** $p(\theta|\alpha)$

$$arg\min_{\theta} \sum_{(y_i,x_i)\in D} \ell(y_i, x_i, \theta) + \ln p(\theta|\alpha)$$

  - Guarantees convergence: $\ln p(\theta|\alpha)$ is a log-barrier function.

- **Unbiased estimates of the expected sufficient statistics**:

$$E_z[s(y_t, z, x_t)|\theta] = \sum_z p(z|y_t, x_t, \theta)s(y_t, z, x_t)$$

  - Collapsed Gibbs sampling is OK!.
  - Variational inference provides unbiased estimates. How sdEM would work?

Some applications of sdEM

- **Text Classification:**
  - Online Discriminative learning of Multinomial NB and LDA.
  - Good results!

Some applications of sdEM

- **Text Classification:**
  - Online Discriminative learning of Multinomial NB and LDA.
  - Good results!
- **Missing Data**:
  - Logistic Regression with missing data

Some applications of sdEM

- **Text Classification:**
  - Online Discriminative learning of Multinomial NB and LDA.
  - Good results!
- **Missing Data**:
  - Logistic Regression with missing data = sdEM + NB + NCLL loss

Some applications of sdEM

- **Text Classification:**
  - Online Discriminative learning of Multinomial NB and LDA.
  - Good results!
- **Missing Data**:
  - Logistic Regression with missing data = sdEM + NB + NCLL loss
  - Linear SVM with missing data

Some applications of sdEM

- **Text Classification:**
  - Online Discriminative learning of Multinomial NB and LDA.
  - Good results!
- **Missing Data**:
  - Logistic Regression with missing data = sdEM + NB + NCLL loss
  - Linear SVM with missing data = sdEM + NB + Hinge loss

Some applications of sdEM

- **Text Classification:**
  - Online Discriminative learning of Multinomial NB and LDA.
  - Good results!
- **Missing Data**:
  - Logistic Regression with missing data = sdEM + NB + NCLL loss
  - Linear SVM with missing data = sdEM + NB + Hinge loss
- **Class Noise**:
  - Generative modeling of the noise.
  - Discriminative performance.

sdEM in AMIDST problems

- **Parameter Learning of TAN models:**
  - Maximum Likelihood = 1 pass over data for counting.
  - Discriminative learning = 1 or 2 pass over data for counting and classifying.

sdEM in AMIDST problems

- **Parameter Learning of TAN models:**
  - Maximum Likelihood = 1 pass over data for counting.
  - Discriminative learning = 1 or 2 pass over data for counting and classifying.

- **Many (sequence,label) pairs:**
  - Set of i.i.d. labelled sequences.
  - $D = \{(y_1, \mathbf{s}_1), ..., (y_1, \mathbf{s}_N)\}$

sdEM in AMIDST problems

- **Parameter Learning of TAN models:**
  - Maximum Likelihood = 1 pass over data for counting.
  - Discriminative learning = 1 or 2 pass over data for counting and classifying.

- **Many (sequence,label) pairs:**
  - Set of i.i.d. labelled sequences.
  - $D = \{(y_1, \mathbf{s}_1), ..., (y_1, \mathbf{s}_N)\}$

- **One Sequence of (element,label) pairs**:
  - Sequence $D = \{(y_1, e_1), ..., (y_T, e_T)\}$
  - No Hidden Variables.
  - Hidden Variables?