Lecture 3.

# MLE and Alternatives

Note Title 2/2/2010

Exponential Distributions, Sufficient Statistics, and the Maximum Entropy Principle.

Most distributions can be expressed in exponential form:

$$P(d|\underline{\lambda}) = \frac{1}{Z[\underline{\lambda}]} e^{\underline{\lambda}\cdot\underline{\phi}(d)}$$

$\underline{\phi}(.)$ is the "sufficient statistics".

This can be derived from the maximum entropy principle:

maximize $-\sum_d p(d)\log p(d) + \underline{\lambda}\cdot\{\sum_d p(d)\underline{\phi}(d) - \underline{\Psi}\}$

"entropy" — Lagrange multiplier — expected stat = observed stats $\Psi$

$\Psi$ - derived from observations see below

Differentiating w.r.t. $p(d)$ gives result.

The value of $\underline{\lambda}$ is chosen s.t. $\sum_d p(d)\underline{\phi}(d) = \underline{\Psi}$.

## Maximum Likelihood Estimation (MLE)

Data. $\{d^\mu : \mu = 1 \text{ to } N\}$

$$p(\{d^\mu\}|\underline{\lambda}) = \prod_\mu P(d^\mu|\underline{\lambda})$$

assumes data is independently identically distributed (i.i.d)

MLE $\hat{\underline{\lambda}} = \arg\min_{\underline{\lambda}} \{-\log p(\{d^\mu\}|\underline{\lambda})\}$

$= \arg\min_{\underline{\lambda}} \{ N \log Z[\underline{\lambda}] - \sum_{\mu=1}^{N} \underline{\lambda}\cdot\underline{\phi}(d^\mu) \}$

$= \arg\min_{\underline{\lambda}} \{ \log Z[\underline{\lambda}] - \underline{\lambda}\cdot\underline{\Psi} \}$ where $\underline{\Psi} = \frac{1}{N}\sum_{\mu=1}^{N} \underline{\phi}(d^\mu)$.

Let. $G[\underline{\lambda}] = \log Z[\underline{\lambda}] - \underline{\lambda}\cdot\underline{\Psi}$.

remark. $\frac{\partial}{\partial\underline{\lambda}} \log Z[\underline{\lambda}] = \sum_d \frac{\underline{\phi}(d) e^{\underline{\lambda}\cdot\underline{\phi}(d)}}{\sum_d e^{\underline{\lambda}\cdot\underline{\phi}(d)}} = \sum_d \underline{\phi}(d) P(d|\underline{\lambda})$

$\frac{\partial^2}{\partial\underline{\lambda}\partial\underline{\lambda}} \log Z[\underline{\lambda}]$ is positive definite, hence $\log Z[\underline{\lambda}] - \underline{\lambda}\cdot\underline{\Psi}$ is convex, unique minimum.

minimum occurs when $\frac{\partial G}{\partial\underline{\lambda}} = 0$

implies $\sum_d \underline{\phi}(d) P(d|\underline{\lambda}) = \underline{\Psi}$ — like maximum entropy

(2)

Note: since $G[\underline{\lambda}]$ is convex it may seem easy to find its global minimum.

But (almost) all algorithms to find it will require computing the gradient

→ eg. steepest descent

$$\underline{\lambda}^{t+1} = \underline{\lambda}^t - \epsilon \left\{ \sum_d \underline{\phi}(d) P(d|\underline{\lambda}^t) + \underline{\Psi} \right\}$$

this requires computing the expectation $\sum_d \underline{\phi}(d) P(d|\underline{\lambda}^t)$ which is difficult because computing the partition function is very hard — except on trees.

Alternative algorithm

Generalized Iterative Scaling (GIS)

is a Discrete Iterative Algorithm (DIA).

~ simple modification of steepest descent, but guaranteed to converge (without needing a step size $\epsilon$).

$$\underline{\lambda}^{t+1} = \underline{\lambda}^t - \log \left\{ \sum_d \underline{\phi}(d) P(d|\underline{\lambda}^t) \right\} + \log \underline{\Psi}$$

(Note: log is taken by components, e.g. $\log(a_1, a_2) = (\log a_1, \log a_2)$)

Note: we can always use MCMC to evaluate $\sum_d \underline{\phi}(d) P(d|\underline{\lambda}^t)$ → doesn't need to know $Z[\underline{\lambda}]$.

Note: suppose we initialize $\underline{\lambda}^0 = \underline{0}$

then $P(d|\underline{\lambda}^0) = U(d)$ uniform distribution.

$$\underline{\lambda}^1 = \log \underline{\Psi} - \log \sum_d U(d) \underline{\phi}(d)$$

For some applications: (i) $\sum_d U(d) \underline{\phi}(d)$ can be calculated

(ii) $\underline{\lambda}^1$ is a good estimate

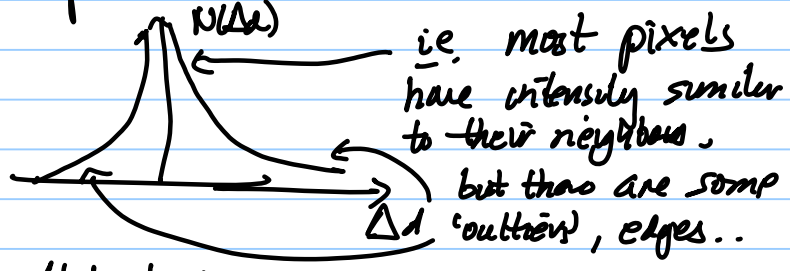
i.e. one iteration of GIS gives a good approximation

## What $\underline{\phi}(.)$'s to Use?

Claim: we can obtain MRF models from stats $\phi(d)$

in particular, we can derive the "weak smoothness" model on MRF's from this approach.

(3) Claim: on most images the histogram of the difference operator $d_{i+1} - d_i$ takes the standard form




i.e. most pixels have intensity similar to their neighbors, but there are some "outliers", edges...

Note: this histogram is not Gaussian

A Gaussian would be too flat at $\Delta d = 0$ and fall away too quickly for large $\Delta d$ ~ doesn't allow for outliers (Gaussians are not "robust").

statistics $\phi(z:d) = \frac{1}{N}\sum_{i=1}^{N} \delta(d_{i+1} - d_i, z)$ histogram

$$\underline{\lambda} \cdot \underline{\phi}(d) = \sum_z \lambda(z) \frac{1}{N} \sum_{i=1}^{N} \delta(d_{i+1} - d_i, z)$$

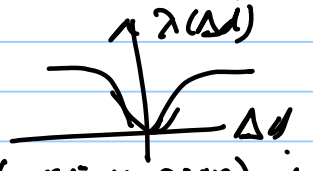
$$= \frac{1}{N} \sum_{i=1}^{N} \left\{ \sum_z \lambda(z)\, \delta(d_{i+1} - d_i, z) \right\}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \lambda(d_{i+1} - d_i) \cdot \; - \frac{1}{N} \sum_{i=1}^{N} \lambda(d_{i+1} - d_i)$$

distribution $p(d|\underline{\lambda}) = \frac{1}{Z[\underline{\lambda}]} e^{\;\;}$

hence an MRF, with potentials $\lambda(\cdot)$ relating neighboring pixels.

How does "weak smoothness" relate to the histogram above?




Answer → one iteration of GIS (previous page) is enough.

What other statistics can we use?

Claim: any 'derivative operator' $\sum_{i=1}^{m} A_i d_i$ s.t. $\sum_{i=1}^{m} A_i$ gives similar statistics (Green)

Applying max entropy / exponential distributions will give MRF's with longer range interactions and higher order cliques.

Statistics can be combined $\underline{\lambda}_1 \cdot \underline{\phi}_1(d) + \underline{\lambda}_2 \cdot \underline{\phi}_2(d)$

$$p(d|\underline{\lambda}_1, \underline{\lambda}_2) = \frac{1}{Z[\underline{\lambda}_1, \underline{\lambda}_2]} e^{\;\;}$$

solve for $\underline{\lambda}_1, \underline{\lambda}_2$ as before

How to select between different combinations?

Model selection → prefer $p(d|\underline{\lambda}_1\underline{\lambda}_2)$ to $p(d|\underline{\lambda})$ if $p(d|\lambda_1,\lambda_2) > p(d|\vec{\lambda})$ for some prior $p(\lambda_1,\lambda_2)\,p(\lambda)$

or, better $\sum_{\lambda_1,\lambda_2} p(d|\lambda_1,\lambda_2)p(\lambda_1,\lambda_2) > \sum_{\lambda} p(d|\lambda)p(\lambda)$

Greedy search Zhu, Wu, Mumford, Della Pietra et al.

(4) Compare MLE to neo machine learning alternatives.

First address simple regression problem

$$p(\omega|d) = \frac{e^{\omega \sum_\mu \lambda_\mu \phi_\mu(d)}}{Z[d,\lambda]}$$

$\omega \in \{\pm 1\}$, implies $Z[d,\lambda] = e^{\sum_\mu \lambda_\mu \phi_\mu(d)} + e^{-\sum_\mu \lambda_\mu \phi_\mu(d)}$

Conditional distribution

Statisticians → treat this as a regression problem.

data $\{(\omega_i, d_i)\}$, maximize $\prod_i p(\omega_i|d_i)$

Note: regression was developed by Gauss around 1800 to predict the motion of a planetoid (Ceres).

(regression can be applied to continuous & discrete variables).

leads to MLE problem:

$$\underset{\text{w.r.t. } \underline{\lambda}}{\text{minimize}} \quad \sum_i \log Z[d_i,\lambda] - \sum_i \omega_i \underline{\lambda}\cdot\underline{\phi}(d_i)$$

$$\sum_i \log\{e^{\underline{\lambda}\cdot\underline{\phi}(d_i)} + e^{-\underline{\lambda}\cdot\underline{\phi}(d_i)}\} - \underline{\lambda}\cdot\sum_i \omega_i \underline{\phi}(d_i).$$

Note → this computation is practical since $Z[d_i;\lambda]$ can be evaluated.

Note: (i) Statisticians usually work with regression models with a limited number of statistics $\phi_\mu(d)$.

(ii) Statisticians usually minimize the function by steepest descent methods → update all the parameters $\underline{\lambda}$ simultaneously.

Contrast with machine learning (AdaBoost)

Changes: (i) call the statistics 'weak classifiers' and require them to be binary-valued.

(ii) minimize a related cost function

$$C_{AdaBoost}(\lambda) = \sum_i e^{\omega_i \sum_\mu \lambda_\mu \phi_\mu(d_i)}$$ → bounds the classification loss

(iii) algorithm: initialize $\lambda_\mu = 0$, $\forall\mu$

Coordinate descent. At each time step, pick the $\lambda_\mu$ which maximizes the decrease in cost.

(others fixed) →

(5) AdaBoost Algorithm.

Initialize $\lambda_\mu = 0$, $\forall \mu$.

At time step $t$: $\{\lambda_\mu^t\}$,

- For each $\mu$, solve $\frac{\partial C_{Ada}(\underline{\lambda})}{\partial \lambda_\mu} = 0$ to solve for $\hat{\lambda}_\mu$
- Calculate $\hat{\mu} = \underset{\mu}{\text{ARGMAX}}\, C_{Ada}(\underline{\lambda}^t + \hat{\lambda}_\mu)$ $\quad (\underline{\lambda}^t + \hat{\lambda}_\mu = (\lambda_1^t, \ldots, \hat{\lambda}_\mu, \ldots, \lambda_D^t))$
- Set $\lambda_{\hat{\mu}}^{t+1} = \hat{\lambda}_{\hat{\mu}}$
  $\lambda_\nu^{t+1} = \lambda_\nu^t$ for $\nu \neq \hat{\mu}$

Note: At time $t$, the algorithm either "selects" a new weak classifier $\phi_\mu(\cdot)$ (s.t. $\lambda_\mu^t = 0$) and assigns it a non-zero "weight" $\lambda_\mu^t$, or it changes the weight of a weak classifier that has already been selected.

→ AdaBoost (i.e. Machine learning) puts much more emphasis on 'selection' → i.e. you start with a dictionary of many weak classifiers $\{\phi_\mu(\cdot) : \mu \in \mathcal{D}\}$ and only select and use a small number (but "small" may mean 1,000's or 10,000's).

→ AdaBoost (Machine learning) emphasizes 'discriminative' rather than learning the conditional distribution $p(w|x)$
  ↝ pays attention to data near the decision boundary and the margins.
  ↝ arguably better than MLE if you have limited data
  ↝ need to cross-validate to prevent over-learning.

→ The two algorithmic steps of AdaBoost
  – i.e. solve $\frac{\partial C_{Ada}}{\partial \lambda_\mu} = 0$, and $\hat{\mu} = \underset{\mu}{\text{ARGMIN}}\, C(\underline{\lambda}^t + \lambda_\mu)$ —
  have simple analytic solutions, because of the special form of $C_{Ada}$.

→ Note: Statisticians and Machine learning researchers are different social communities.

(6) Comparisons.

MLE: log-likelihood
$$\min_{\underline{\lambda}} \left\{ \sum_i \log\left(e^{\underline{\lambda}\cdot\underline{\phi}(d_i)} + e^{-\underline{\lambda}\cdot\underline{\phi}(d_i)}\right) - \underline{\lambda}\cdot\sum_i \omega_i \underline{\phi}(d_i) \right\}$$

AdaBoost:
$$\min_{\underline{\lambda}} \left\{ \sum_i e^{\omega_i \underline{\lambda}\cdot\underline{\phi}(d_i)} \right\}$$

Max-Margin ~SVM
$$\min_{\underline{\lambda}} \left\{ \frac{1}{2}|\underline{\lambda}|^2 + C \sum_i \max\{0, 1-\omega_i \underline{\lambda}\cdot\underline{\phi}(d_i)\} \right\}$$
'regularizer' "hinge loss"

Also motivated by Machine learning margin ideas.

Also, like AdaBoost, designed to enable efficient learning algorithms.

Optimization Techniques:
(1) Steepest Descent / Discrete Iterative Algorithms.
(2) Coordinate Descent (e.g. AdaBoost)
(3) Online Learning.
→ select a sample $x_i$ at random,
→ do iteration of steepest descent.

High level Points.

- The selection of features is easier for AdaBoost than for learning MRF (Della Pietra et al, Zhu et al) Because AdaBoost only learns a distribution on one variable $\omega$. AND AdaBoost uses a simpler cost function.