Lecture 9. Machine Learning: Structure & Latents

Note Title 2/4/2010

Structure Max-Margin extends binary-classification methods so they can be applied to learn the parameters of an MRF, HMM, SCFG or other model.

Recall standard SVM for binary classification.

$$R(\underline{\lambda}) = \tfrac{1}{2}|\underline{\lambda}|^2 + C\sum_{i=1}^{n} \max\{0, 1 - y_i\, \underline{\lambda}\cdot\underline{\phi}(\underline{d}_i)\}$$

Training Data $\{(y_i, \underline{d}_i)\}$ $\quad y_i \in \{\pm 1\}$

e.g. to get a plane, set $\underline{\phi}(\underline{d}) = \underline{d}$.

Decision rule: $\hat{y}_i(\underline{\lambda}) = \arg\max_y y\, \underline{\lambda}\cdot\underline{\phi}(\underline{d}_i)$
$= \operatorname{sgn}\ \underline{\lambda}\cdot\underline{\phi}(\underline{d})$.

The task is to minimize $R(\underline{\lambda})$ w.r.t. $\underline{\lambda}$ which maximizes the 'margin' $1/|\underline{\lambda}|$.

Here is a more general formulation that can be used if the output variable y is a vector $y = (y_1 \dots y_w)$ — i.e. it could be the state of an MRF, an HMM, or a SCFG.

$$R(\underline{\lambda}) = \tfrac{1}{2}|\underline{\lambda}|^2 + C\sum_{i=1}^{n} \Delta(\underline{y}_i, \hat{\underline{y}}_i(\underline{\lambda}))$$

decision rule: $\hat{\underline{y}}_i(\underline{\lambda}) = \arg\max_{\underline{y}} \underline{\lambda}\cdot\underline{\Phi}(\underline{d}, \underline{y})$.

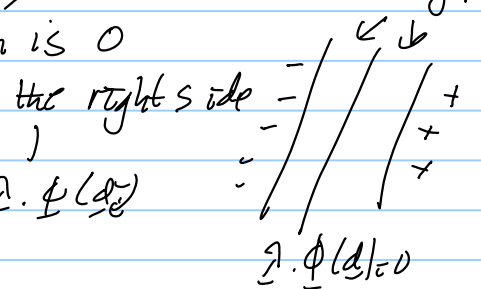
the error function $\Delta(\underline{y}_i, \hat{\underline{y}}_i(\underline{\lambda}))$ is any measure of distance between the true solution $\underline{y}_i$ and the estimate $\hat{\underline{y}}_i(\underline{\lambda})$

→ to obtain binary-value. → (i) set $\underline{y}_i = y_i \in \{-1, 1\}$

(ii) $\underline{\Phi}(\underline{d}, \underline{y}) = y\,\underline{\Phi}(\underline{d})$

(iii) $\Delta(y_i, \hat{y}_i(\underline{\lambda})) = \max\{0, 1 - y_i\, \underline{\lambda}\cdot\underline{\phi}(\underline{d}_i)\}$

Hinge loss ——✓ because the function is 0 if $y_i\, \underline{\lambda}\cdot\underline{\phi}(\underline{d}_i) > 1$ (i.e. point is on the right side of the margin) and the function increases linearly with $\underline{\lambda}\cdot\underline{\phi}(\underline{d}_i)$



(iv) $\hat{y}_i(\underline{\lambda}) = \arg\max_y y\, \underline{\lambda}\cdot\underline{\phi}(\underline{d})$

(2) This more general formulation is $P(y,d) = \frac{1}{Z} e^{\underline{\lambda}.\underline{\phi}(y,d)}$

$$R(\underline{\lambda}) = \tfrac{1}{2}|\underline{\lambda}|^2 + C \sum_{i=1}^{m} \Delta(\underline{y}_i : \hat{\underline{y}}_i(\underline{\lambda}))$$

$$\hat{\underline{y}}_i = \arg\max_{\underline{y}} \underline{\lambda}.\underline{\phi}(\underline{y},\underline{d})$$

(*) This requires an inference algorithm, like in the last few lectures.

for binary classification inference only has to compute $\max_{y_i \in \{\pm 1\}} y_i \underline{\lambda}.\underline{\phi}(\underline{d})$ so is trivial.

(*) Also need to be able to minimize $R(\underline{\lambda})$ to find $\underline{\lambda}$ → hard because the error term $\Delta(\underline{y}_i : \hat{\underline{y}}_i(\underline{\lambda}))$ is a highly complicated function of $\underline{\lambda}$.

Modify $R(\underline{\lambda})$ to an upper bound $\bar{R}(\underline{\lambda})$

$$\bar{R}(\underline{\lambda}) = \tfrac{1}{2}|\underline{\lambda}|^2 + C \sum_{i=1}^{m} \{ \max_{\underline{y}} \{ \Delta(\underline{y}_i;\underline{y}) + \underline{\lambda}.\underline{\phi}(\underline{d}_i,\underline{y}) \} - \underline{\lambda}.\underline{\phi}(\underline{d}_i,\underline{y}_i) \}$$

which is convex in $\underline{\lambda}$. hence has a single minimum.

To get this bounds use two steps:

(step 1) $$\max_{\underline{y}} \{ \Delta(\underline{y}_i;\underline{y}) + \underline{\lambda}.\underline{\phi}(\underline{d}_i,\underline{y}) \} \geq \Delta(\underline{y}_i;\hat{\underline{y}}_i(\underline{\lambda})) + \underline{\lambda}.\underline{\phi}(\underline{d}_i,\hat{\underline{y}}_i(\underline{\lambda}))$$

→ $\hat{\underline{y}}_i(\underline{\lambda})$ maximizes $\underline{\lambda}.\underline{\phi}$

→ equality if it also maximizes $\Delta + \underline{\lambda}.\underline{\phi}$

(step 2) $$\underline{\lambda}.\underline{\phi}(\underline{d}_i,\hat{\underline{y}}_i(\underline{\lambda})) \geq \underline{\lambda}.\underline{\phi}(\underline{d}_i,\underline{y}_i)$$

Note: bounds are 'tight' because if we can find a good solution then $\underline{y}_i \approx \hat{\underline{y}}_i(\underline{\lambda})$.

How to minimize $\bar{R}(\underline{\lambda})$?

Several Algorithms (hot topic)

Some in dual space — like original SVM for binary problems.

Simple: stochastic gradient descent

pick example $(\underline{d}_i, \underline{y}_i)$

take derivative of $\bar{R}(\underline{\lambda})$ w.r.t. $\underline{\lambda}$.

$$\underline{\lambda}^{t+1} = \underline{\lambda}^t - \epsilon C \{ \underline{\phi}(\underline{d}_i,\hat{\underline{y}}) - \underline{\phi}(\underline{d}_i,\underline{y}_i) \}$$

where $\hat{\underline{y}} = \text{ARGMAX} \{ \Delta(\underline{y}_i;\underline{y}) + \underline{\lambda}.\underline{\phi}(\underline{x}_i,\underline{y}) \}$

Note: inference algorithm must be adapted to compute this.

(3) How to extend to models with latent (hidden) variables? Denote these variables by $\underline{h}$.

Want decision rule $(\hat{\underline{y}}, \hat{\underline{h}}) = \arg\max_{(\underline{y},\underline{h}) \in \mathcal{Y} \times \mathcal{H}} \underline{\lambda} \cdot \underline{\phi}(\underline{d}, \underline{y}, \underline{h})$ ← must be computable by inference algorithm

Training data $\{(\underline{d}^i, \underline{y}^i) : i = 1, \dots N\}$. the hidden variables are not known.

Loss function $\Delta(\underline{y}_i; \hat{\underline{y}}_i(\underline{\lambda}), \hat{\underline{h}}_i(\underline{\lambda}))$

depends on the truth $\underline{y}_i$

the estimate of $\hat{\underline{y}}_i(\underline{\lambda}), \hat{\underline{h}}_i(\underline{\lambda})$ from the model

$$R(\underline{\lambda}) = \tfrac{1}{2}|\underline{\lambda}|^2 + C \sum_{i=1}^{m} \Delta(\underline{y}_i : \hat{\underline{y}}_i(\underline{\lambda}), \hat{\underline{h}}_i(\underline{\lambda}))$$

non-trivial function of $\underline{\lambda}$

replace $R(\underline{\lambda})$ by

$$\tilde{R}(\underline{\lambda}) = \tfrac{1}{2}|\underline{\lambda}|^2 + C \sum_{i=1}^{m} \Big\{ \max_{(\underline{y},\underline{h})} \{ \Delta(\underline{y}_i; \underline{y}, \underline{h}) + \underline{\lambda} \cdot \underline{\phi}(\underline{d}_i; \underline{y}, \underline{h}) \} - \max_{\underline{h}} \underline{\lambda} \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i, \underline{h}) \Big\}$$

$$\tilde{R}(\underline{\lambda}) = f(\underline{\lambda}) + g(\underline{\lambda}), \quad \text{with } g(\underline{\lambda}) = -\max_{\underline{h}} \underline{\lambda} \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i, \underline{h})$$

f: convex, g: concave

To show convexity and concavity

suppose $\tau(\underline{\lambda}) = \sum_{i=1}^{n} \max_{\underline{y}_i} \underline{\lambda} \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i)$

convex if $\tau(\alpha \underline{\lambda}_1 + (1-\alpha)\underline{\lambda}_2) \le \alpha \tau(\underline{\lambda}_1) + (1-\alpha)\tau(\underline{\lambda}_2)$

$$\tau(\alpha \underline{\lambda}_1 + (1-\alpha)\underline{\lambda}_2) = \sum_{i=1}^{n} \max_{\underline{y}_i} \{ (\alpha \underline{\lambda}_1 + (1-\alpha)\underline{\lambda}_2) \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i) \}$$

$$\alpha \tau(\underline{\lambda}_1) + (1-\alpha)\tau(\underline{\lambda}_2) = \alpha \sum_{i=1}^{n} \max_{\underline{y}_i} \underline{\lambda}_1 \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i) + (1-\alpha) \sum_{i=1}^{n} \max_{\underline{y}_i} \underline{\lambda}_2 \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i)$$

but $\max_{\underline{y}_i} \alpha \underline{\lambda}_1 \cdot \underline{\phi}(\underline{x}_i, \underline{y}_i) + \max_{\underline{y}_i} (1-\alpha) \underline{\lambda}_2 \cdot \underline{\phi}(\underline{x}_i, \underline{y}_i)$

$$\ge \max_{\underline{y}_i} \{ (\alpha \underline{\lambda}_1 + (1-\alpha)\underline{\lambda}_2) \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i)$$

hence $f(\cdot)$ is convex

and $g(\cdot)$ is concave.

(4) Apply CCCP algorithm.

Two stages: step 1.

$$\frac{\partial}{\partial \underline{\lambda}} g(\underline{\lambda}^t) = -\underline{\phi}(\underline{d}_i, \underline{y}_i, \underline{h}^t)$$

where $\underline{h}^* = \arg\max_{\underline{h}} \underline{\lambda}^t \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i, \underline{h})$

$\underline{\lambda}^t$ current estimate of $\underline{\lambda}$.

step 2. solve

$$\underline{\lambda}^{t+1} = \arg\min_{\underline{\lambda}} \left(f(\underline{\lambda}) + \underline{\lambda} \cdot \frac{\partial}{\partial \underline{\lambda}} g(\underline{\lambda}^t) \right)$$

This reduces to a modified SVM with known state

$$\min_{\underline{\lambda}} \frac{1}{2}|\underline{\lambda}|^2 + C \sum_{i=1}^{n} \max_{(\underline{y},\underline{h})} \{ \underline{\lambda} \cdot \underline{\phi}(\underline{d}_i, \underline{y}, \underline{h}) + \Delta(\underline{y}_i, \underline{y}, \underline{h}) \} - C \sum_{i=1}^{n} \underline{\lambda} \cdot \underline{\phi}(\underline{d}_i, \underline{y}_i, \underline{h}_i^*)$$

Note: similarities to EM.

→ step 1 involves estimating the hidden state $\underline{h}_i^*$

→ step 2 estimates $\underline{\lambda}$

repeat.

Note: like EM there is no guarantee that this will converge to the global optimum.