(1) # Multi-Class Multi-~~State~~ SVM

Note Title 11/21/2011

$$L_p(\underline{\omega}, z; \alpha) = \frac{1}{2}|\underline{\omega}|^2 + C\sum_i z_i - \sum_i \sum_y \alpha_y^i \{ z_i - \ell(y, y_i) - \underline{\omega}\cdot\underline{\phi}(\underline{x}_i : \underline{y}) + \underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}_i)\}$$

solution $\hat{\underline{y}}(\underline{x} : \underline{\omega}) = \arg\max_{\underline{y}} \underline{\omega}\cdot\underline{\phi}(\underline{x} : \underline{y})$

Constraint
$$z_i - \ell(y, y_i) - \underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}) + \underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}_i) \geq 0$$
$$z_i \geq \max_{y} \{ \ell(\underline{y}, \underline{y}_i) + \underline{\omega}\cdot\underline{\phi}(\underline{x}_i : \underline{y}) - \underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}_i)\}$$

$$\frac{1}{2}|\underline{\omega}|^2 + C \sum_i \max_y \{ \ell(\underline{y}, \underline{y}_i) + \underline{\omega}\cdot\underline{\phi}(\underline{x}_i : \underline{y}) - \underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}_i)\}.$$

Note: no need to separately impose $z_i \geq 0$ because if we set $\underline{y} = \underline{y}_i$, we see $z_i \geq \ell(y_i, y_i) + \cancel{\underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}_i)} - \cancel{\underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}_i)}$ $= 0$

Solve the primal problem
$$\frac{\partial}{\partial \underline{\omega}} L_p(\underline{\omega}, z : \alpha) = 0$$
$$\Rightarrow \underline{\omega} = \sum_i \sum_y \alpha_y^i \{ \underline{\phi}(\underline{x}_i, \underline{y}_i) - \underline{\phi}(\underline{x}_i, \underline{y})\}$$
like support vectors

note: if $\underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}_i) > \underline{\omega}\cdot\underline{\phi}(\underline{x}_i, \underline{y}) + \ell(\underline{y}_i, \underline{y})$ then $\alpha_y^i = 0$

$$\frac{\partial}{\partial z_i} L_p(\underline{\omega}, z; \alpha) = 0$$
$$\Rightarrow C = \sum_y \alpha_y^i$$
so $\frac{\alpha_y^i}{C}$ is a probability distribution.

This gives solution $\underline{\omega}(\alpha), z(\alpha)$

Substituting back gives the dual energy:
$$L_d(\underline{\alpha}) = L_p(\underline{\omega}(\alpha), z(\alpha); \underline{\alpha})$$
$$= \sum_i \sum_y \alpha_y^i \ell(y, y_i) - \frac{1}{2} \sum_{i,j} \sum_{y,z} \alpha_y^i \alpha_z^i \{ \underline{\phi}(\underline{x}_i, \underline{y}_i) - \underline{\phi}(\underline{x}_i, \underline{y})\} \cdot \{ \underline{\phi}(\underline{x}_j, \underline{y}_j) - \underline{\phi}(\underline{x}_j, \underline{z})\}$$

The case of binary classification can be recovered by setting $\ell(y, y_i) = 1$, if $y \neq y_i$
$= 0$, if $y = y_i$

$$\underline{\phi}(\underline{x}, y) = y\, \underline{\phi}(\underline{x})$$

(2) Example (A)

Face or Not face $\rightarrow$ $\boxed{y_1}$ $\quad$ $\boxed{y_2}$ $\leftarrow$ Face or Not face

$\underline{x}_1 \qquad \underline{x}_2$

$y_1 \in \{\pm 1\}, \qquad y_2 \in \{\pm 1\},$

For each window separately $\underline{\phi}(\underline{x})$ set of Haar filters (like AdaBoost)

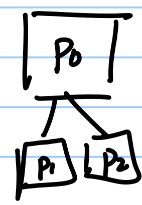
Together. $\quad \underline{y} = (y_1, y_2) \quad \underline{x} = (x_1, x_2)$

$$\underline{\psi}(y_1,y_2) = (\delta_{y_1,1}\,\delta_{y_2,1},\ \delta_{y_1,1}\,\delta_{y_2,0},\ \delta_{y_1,0}\,\delta_{y_2,1},\ \delta_{y_1,0}\,\delta_{y_2,0})$$

$$\underline{\omega}\cdot\underline{\phi}(\underline{y},\underline{x}) = y^1\,\underline{\omega}^1\cdot\underline{\phi}(\underline{x}^1) + y^2\,\underline{\omega}^2\cdot\underline{\phi}(\underline{x}^2) + \underline{\omega}^{12}\cdot\underline{\psi}(y_1,y_2).$$

Example (B)

$\underline{h} = (p_0, p_1, p_2)$



$y = 1 \qquad y = -1$

$\underline{\lambda}\cdot\underline{\varphi}(\underline{x},\underline{h},y=-1) = 0$

$\underline{\lambda}\cdot\underline{\varphi}(\underline{x},\underline{h},y=1) \qquad \underline{h} = p_0, p_1, p_2$

$$= \underline{\lambda}\cdot\underline{\varphi}(\underline{x}, p_0, p_1, p_2, y=1)$$
$$= \underline{\lambda}^0_1\cdot\{(p_0-p_1), (p_0-p_1)^2\}$$
$$+ \underline{\lambda}^0_2\cdot\{(p_0-p_2), (p_0-p_2)^2\}$$
$$+ \underline{\mu}^0\cdot\underline{\varphi}(\underline{x}, p_0)$$
$$+ \underline{\mu}^1\cdot\underline{\varphi}(\underline{x}, p_1)$$
$$+ \underline{\mu}^2\cdot\underline{\varphi}(\underline{x}, p_2)$$

During learning the position p_0, p_1, p_2 of the object parts are unspecified

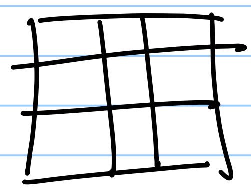
$\underline{x} = \{x_{i,j} : i,j \in D\}$

$\underline{x}^P = \{x_{i,j} : i,j \in D^P\}$

D is the image lattice

$x_{i,j}$ is the intensity at lattice site ij

D^P is a subregion of D.



filters $\quad D^P = \bigcup_{\alpha=1}^{m^2} D_\alpha$

$\underline{f} = (f_1 \ldots f_{m^2})$

D^P

$f_i = f(\underline{x}_{D^P_i})$

$= f\{x_{ij} : (i,j) \in D^P_\alpha\}$

Quantize the response values

$f(\underline{x}_{D^P}) \in \{\alpha : \alpha \in \Lambda\}$

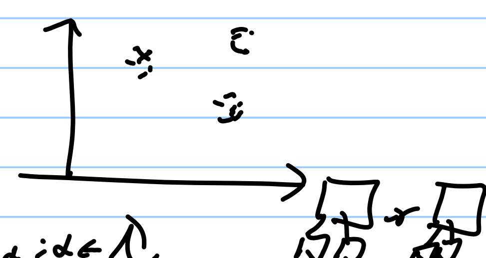
Total response: histogram $\quad n_\alpha = \sum_a \delta_{f(\underline{x}_{D^P}),\alpha}.$

Quantize

cluster the set of values $\{f_{\alpha}\}$

use k-means

each datapoint (feature value) is associated to the closest mean $\{m_\alpha : \alpha \in \Lambda\}$



Extension, to multiple models (to allow for different viewpoints)

$\underline{\lambda}\cdot\underline{\varphi}(\underline{x}, \mu, p^\mu_0, p^\mu_1, p^\mu_2, y=1) \qquad \mu$ label models

Lecture 9. Machine Learning: Structure & Latent.

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.

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

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: $\hat{y}_i(\underline{\lambda}) = \arg\max_{y} y\, \underline{\lambda}\cdot\underline{\phi}(d_i)$
$= \text{sgn}\ \underline{\lambda}\cdot\underline{\phi}(d_i)$.

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 $y = (y_1 \cdots y_N)$. — i.e. it could be the state of an MRF, an HMM, or a SCFG.

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

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

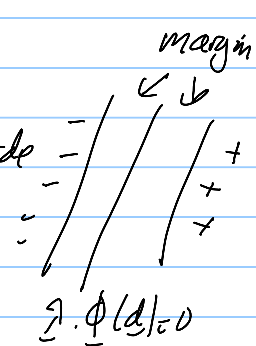
the error function $\Delta(\underline{y}_i; \hat{\underline{y}}_i(\underline{\lambda}))$ is any measure of distance between the true solution $\underline{y}_i$ and the estimate $\hat{\underline{y}}_i(\underline{\lambda})$.

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

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

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

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 $P(\underline{y}, \underline{d}) = \frac{1}{Z} e^{\underline{\lambda} \cdot \underline{\phi}(\underline{y}, \underline{d})}$

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

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

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

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

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

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

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

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

To get this bounds use two steps:

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

$\rightarrow$ $\underline{\hat{y}}_i(\underline{\lambda})$ maximizes $\underline{\lambda} \cdot \underline{\phi}$
$\rightarrow$ equality if it also maximizes $\Delta + \underline{\lambda} \cdot \underline{\phi}$

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

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

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. $\underline{\lambda}$.

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

where $\underline{\hat{y}} = \mathrm{ARGMAX} \{ \Delta(\underline{y}_i, \underline{y}) + \underline{\lambda} \cdot \underline{\phi}(\underline{x}_i, \underline{y}) \}$

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

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

Training data $\{(\underline{d}^i, \underline{y}^i) : i = 1 \text{ to } N\}$. 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

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

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

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

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

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

To show convexity and concavity

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

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

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

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

but

$$\max_{\hat{\underline{y}}_i} \alpha \underline{\lambda}_1 \cdot \underline{\phi}(\underline{x}_i, \hat{\underline{y}}_i) + \max_{\hat{\underline{y}}_i} (1-\alpha)\underline{\lambda}_2 \cdot \underline{\phi}(\underline{x}_i, \hat{\underline{y}}_i) \geq \max_{\hat{\underline{y}}_i} \{(\alpha \underline{\lambda}_1 + (1-\alpha)\underline{\lambda}_2) \cdot \underline{\phi}(\underline{d}^i ; \hat{\underline{y}}_i)\}$$

hence $f(.)$ is convex

and $g(.)$ is concave.

(4) <u>Apply</u> CCCP algorithm.

<u>Two stages</u>: <u>step 1</u>. $\frac{\partial g(\underline{\lambda})}{\partial \underline{\lambda}}^{t} = -\underline{\phi}(\underline{d}_i, \underline{y}_i, \underline{h}^*)$ 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 $\underline{\lambda}$.

<u>step 2</u>. 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

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

<u>Note</u>: similarities to EM.

→ step 1 involves estimating the hidden state h_i^*

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

repeat.

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