

A majorization-minimization algorithm for (multiple) hyperparameter learning

Chuan-Sheng Foo Chuong B. Do Andrew Y. Ng

Stanford University

ICML 2009

Montreal, Canada

17th June 2009

Supervised learning

- Training set of m IID examples

$$\mathcal{D} = \left\{ (x^{(i)}, y^{(i)}) \right\}_{i=1}^m$$

Labels may be real-valued, discrete, structured

- Probabilistic model $p(y|x; \mathbf{w})$
- Estimate parameters $\mathbf{w} \in \mathbb{R}^n$

Regularization prevents overfitting

- Regularized maximum likelihood estimation

L_2 -regularized Logistic Regression

Regularization Hyperparameter

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \left[\sum_{i=1}^m \log \left(1 + \exp \left(-y^{(i)} \mathbf{w}^T \mathbf{x}^{(i)} \right) \right) + \frac{1}{2} C \|\mathbf{w}\|^2 \right]$$

- Also maximum *a posteriori* (MAP) estimation

$$\mathbf{w}_{MAP} = \arg \min_{\mathbf{w}} \left[\underbrace{-\log p(\mathcal{D}|\mathbf{w})}_{\text{Data log-likelihood}} - \underbrace{\log p(\mathbf{w}; C)}_{\text{Log-prior over model parameters}} \right]$$

Data log-likelihood

Log-prior over model parameters

How to select the hyperparameter(s)?

- Grid search
 - + Simple to implement
 - Scales exponentially with # hyperparameters
- Gradient-based algorithms
 - + Scales well with # hyperparameters
 - Non-trivial to implement

Can we get the best of both worlds?

Our contribution

- ✓ **Striking ease of implementation**
 - ✓ Simple, closed-form updates for C
 - ✓ Leverage existing solvers
- ✓ Scales well to multiple hyperparameter case
- ✓ Applicable to wide range of models


Outline

1. Problem definition
2. The “integrate out” strategy
3. The Majorization-Minimization algorithm
4. Experiments
5. Discussion

The “integrate out” strategy

- Treat hyperparameter C as a random variable
- Analytically integrate out C
- Need a convenient prior $p(C)$

$$\arg \min_{\mathbf{w}} \left[-\log p(\mathcal{D}|\mathbf{w}) - \log \underline{p(\mathbf{w}; C)} \right]$$

$$\arg \min_{\mathbf{w}} \left[-\log p(\mathcal{D}|\mathbf{w}) - \log \int_C \underline{p(\mathbf{w}|C)p(C)dC} \right]$$


Integrating out a single hyperparameter

- For L_2 regularization,

$$p(\mathbf{w}|C) \propto \exp\left(-\frac{1}{2}C \|\mathbf{w}\|^2\right)$$

- A convenient prior:

$$C \sim \text{Gamma}(\alpha, \beta)$$

- The result:

$$\log p(\mathbf{w}) = -\left(\frac{n}{2} + \alpha\right) \log\left(\frac{1}{2} \|\mathbf{w}\|^2 + \beta\right)$$

1. C is gone

2. Neither convex nor concave in \mathbf{w}

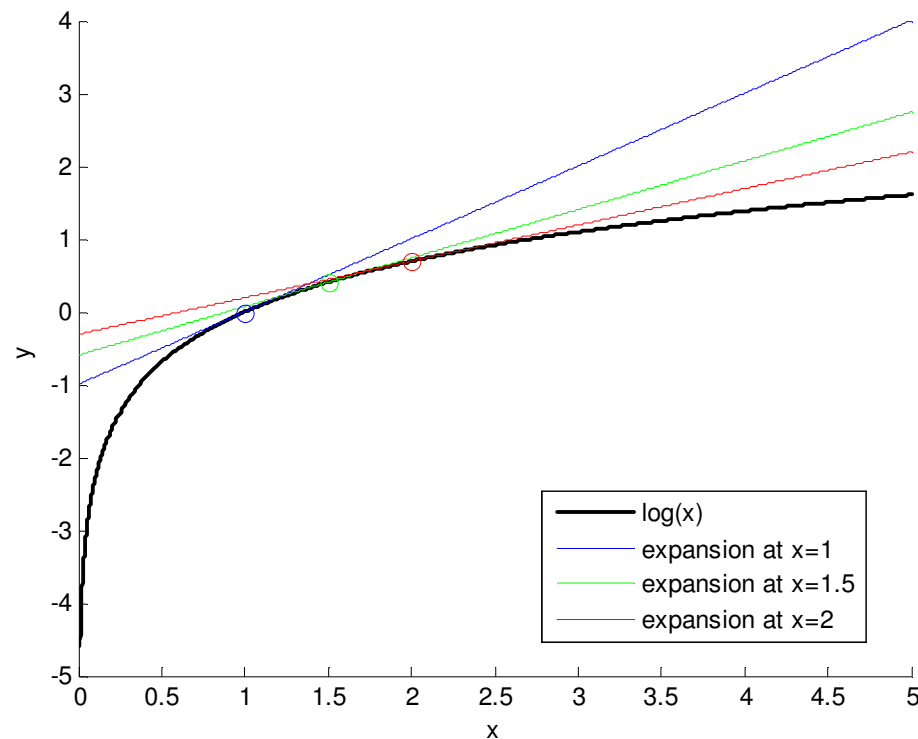
The **M**ajorization-**M**inimization Algorithm

- Replace hard problem by series of easier ones
- EM-like; two steps:
 1. *Majorization*
Upper bound the objective function
 2. *Minimization*
Minimize the upper bound

MM1: Upper-bounding the new prior

- New prior: $\log p(\mathbf{w}) = -\left(\frac{n}{2} + \alpha\right) \log\left(\frac{1}{2} \|\mathbf{w}\|^2 + \beta\right)$
- Linearize the log:

$$\forall x, y \in (0, \infty) \quad \log x \leq \log y + (x - y)/y$$



MM2: Solving the resultant optimization problem

- Resultant linearized prior

$$\left(\frac{n}{2} + \alpha\right) \left[\log \left(\frac{1}{2} \|\mathbf{w}^{(t)}\|^2 + \beta \right) + \frac{\frac{1}{2} \|\mathbf{w}\|^2 + \beta}{\frac{1}{2} \|\mathbf{w}^{(t)}\|^2 + \beta} - 1 \right]$$

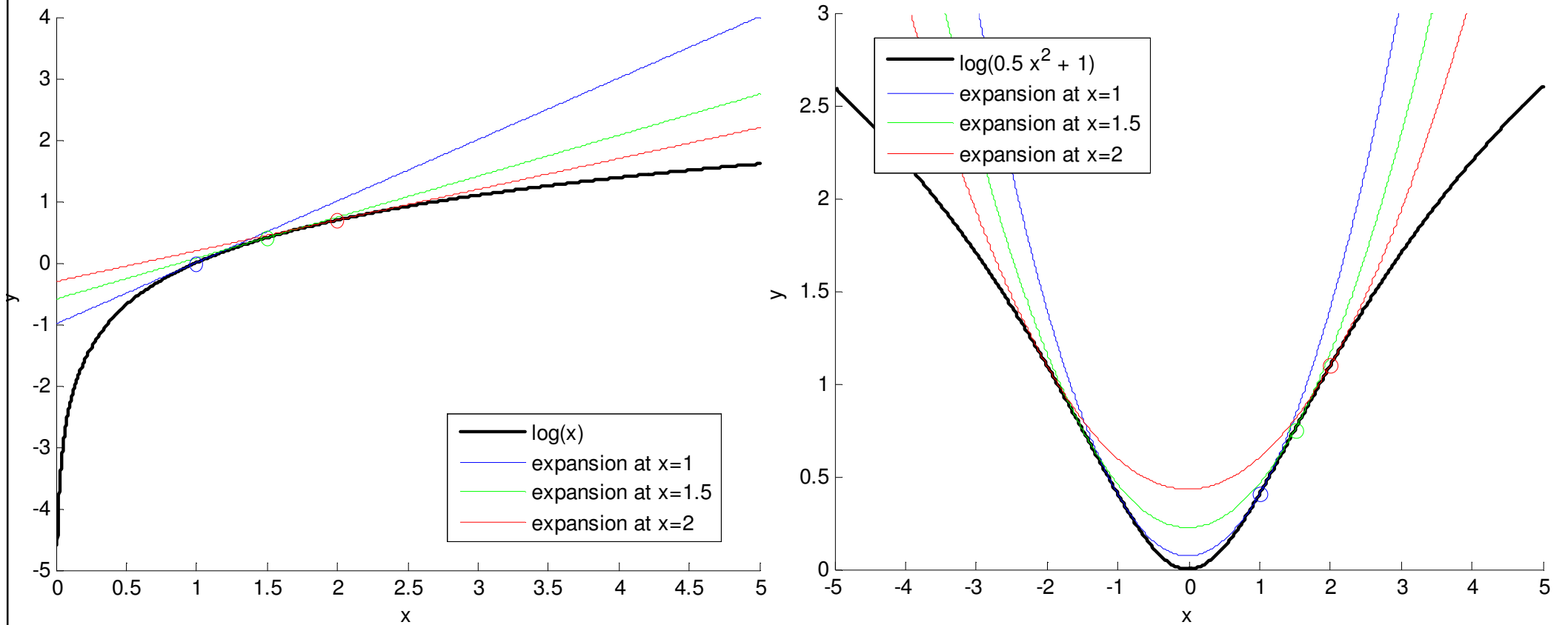
Terms independent of \mathbf{w}

- Get standard L_2 -regularization!

$$\mathbf{w}^{(t+1)} := \arg \min_{\mathbf{w}} \left[-\log p(\mathcal{D}|\mathbf{w}) + \left(\frac{n/2 + \alpha}{\frac{1}{2} \|\mathbf{w}^{(t)}\|^2 + \beta} \right) \left(\frac{1}{2} \|\mathbf{w}\|^2 \right) \right]$$

Use existing solvers!

Visualization of the upper bound



Overall algorithm

Algorithm 1 Majorization-minimization algorithm for single hyperparameter learning

Initialize $t := 0$.

repeat

$$\text{Compute } C^{(t)} := \begin{cases} 1 & \text{if } t = 0 \\ \frac{n/2 + \alpha}{\frac{1}{2} \|\mathbf{w}^{(t)}\|^2 + \beta} & \text{otherwise.} \end{cases}$$

$$\mathbf{w}^{(t+1)} := \arg \min_{\mathbf{w}} \left[-\log p(\mathcal{D}|\mathbf{w}) + \frac{1}{2} C^{(t)} \|\mathbf{w}\|^2 \right].$$

until convergence

1. Closed form updates for C

2. Leverage existing solvers

Converges to a local minimum

What about multiple hyperparameters?

- Regularization groups

$$\mathbf{W} = (\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5)$$

“To C or not to C. That is the question...”

NLP

Unigram
feature
weights

Bigram
feature
weights

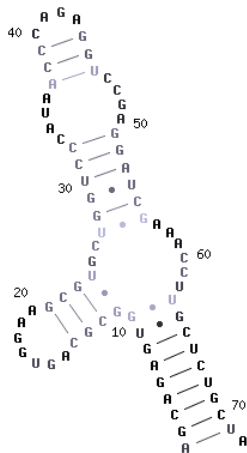
RNA

**Secondary
Structure
Prediction**

Hairpin
loops

Bulge
loops

Mapping from
weights to groups



$$\mathbf{C} = (\mathbf{C}_1 , \mathbf{C}_2)$$

What about multiple hyperparameters?

Algorithm 2 Majorization-minimization algorithm for multiple hyperparameter learning

Initialize $t := 0$.

repeat

For $j = 1, \dots, k$, compute

Separately update each regularization group

Sum weights in each group

$$C_j^{(t)} := \begin{cases} 1 & \text{if } t = 0 \\ \frac{n_j/2 + \alpha}{\frac{1}{2} \sum_{i \in \pi^{-1}(j)} (w_i^{(t)})^2 + \beta} & \text{otherwise.} \end{cases}$$

$$\mathbf{w}^{(t+1)} := \arg \min_{\mathbf{w}} \left[-\log p(\mathcal{D} | \mathbf{w}) + \frac{1}{2} \sum_{i=1}^n C_{\pi(i)}^{(t)} w_i^2 \right].$$

until convergence

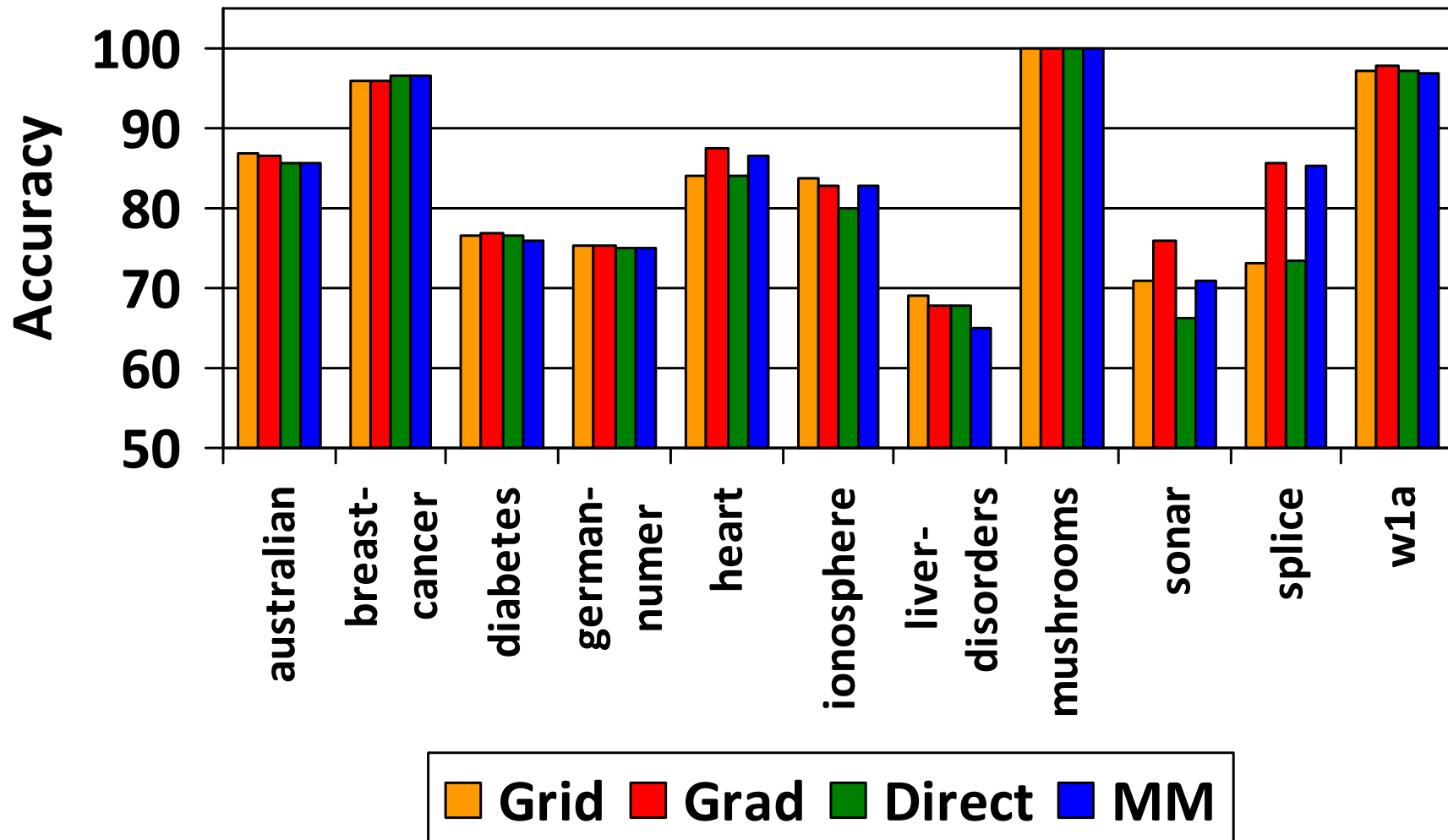
Weighted L_2 -regularization

Experiments

- 4 probabilistic models
 - Linear regression (too easy, not shown)
 - Binary logistic regression
 - Multinomial logistic regression
 - Conditional log-linear model
- 3 competing algorithms
 - Grid search
 - Gradient-based algorithm (Do et al., 2007)
 - Direct optimization of new objective
- Algorithm run with $\alpha = 0$, $\beta = 1$

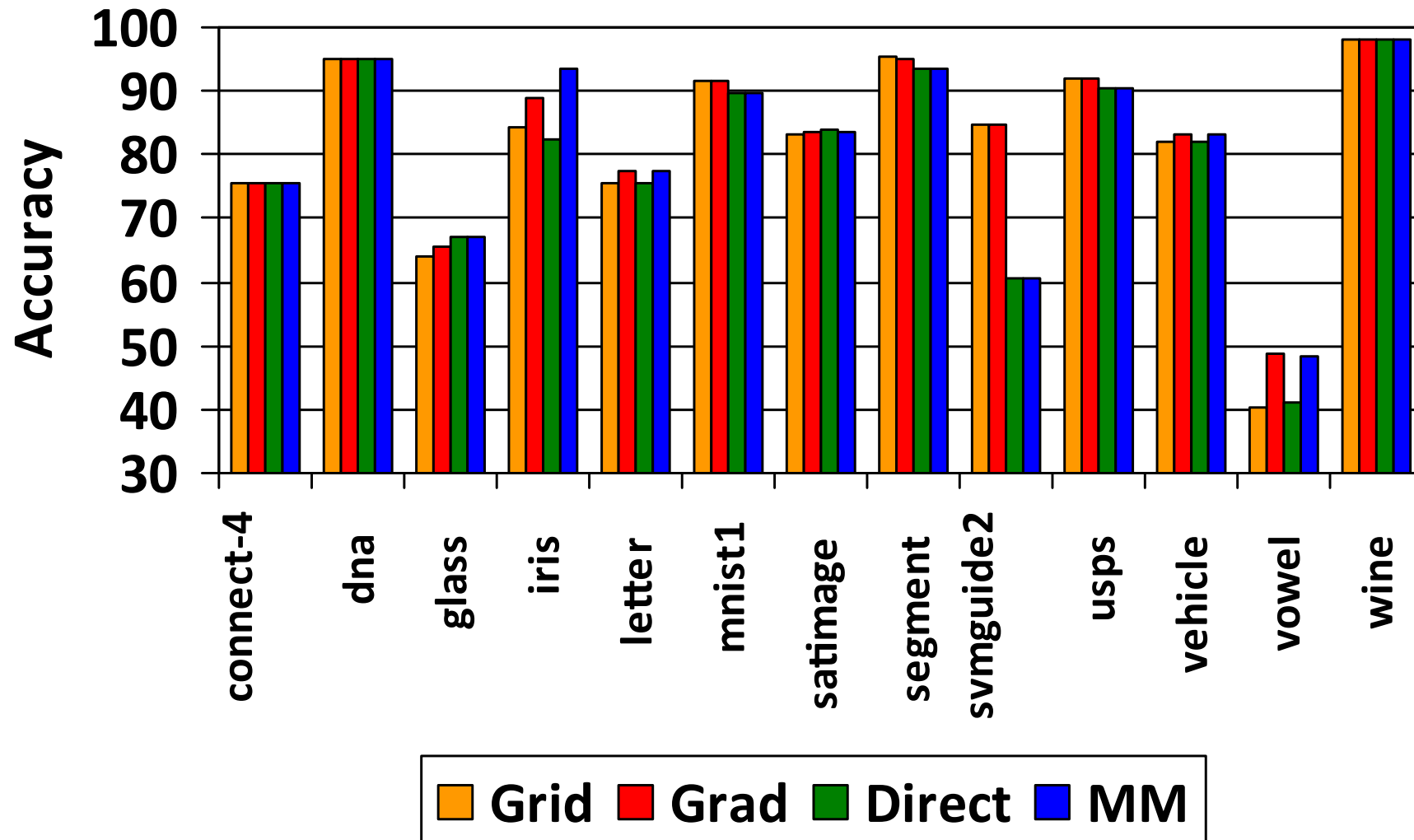
Results: Binary Logistic Regression

$$p(y|\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-y\mathbf{w}^T \mathbf{x})} \quad y \in \{-1, +1\}$$



Results: Multinomial Logistic Regression

$$p(y = k | \mathbf{x}; \mathbf{w}) = \frac{\exp(\mathbf{w}_k^T \mathbf{x})}{\sum_{i=1}^k \exp(\mathbf{w}_i^T \mathbf{x})} \quad y \in \{1, \dots, k\}$$



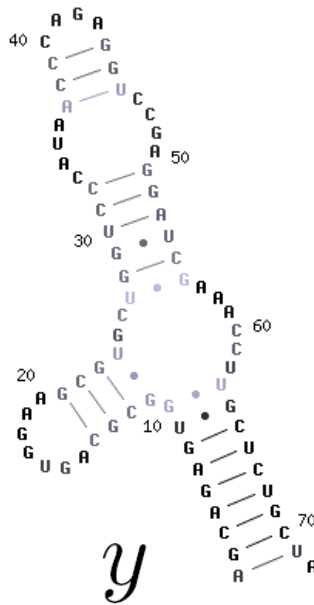
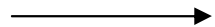
Results: Conditional Log-Linear Models

$$p(y|x; \mathbf{w}) \propto \exp(\mathbf{w}^T \mathbf{F}(x, y))$$

- RNA secondary structure prediction
- Multiple hyperparameters

```
AGCAGAGUGGCGCA
GUGGAAGCGUGCUG
GUCCAUAACCCAGA
GGUCCGAGGAUCGA
AACCUUGCUCUGCUA
```

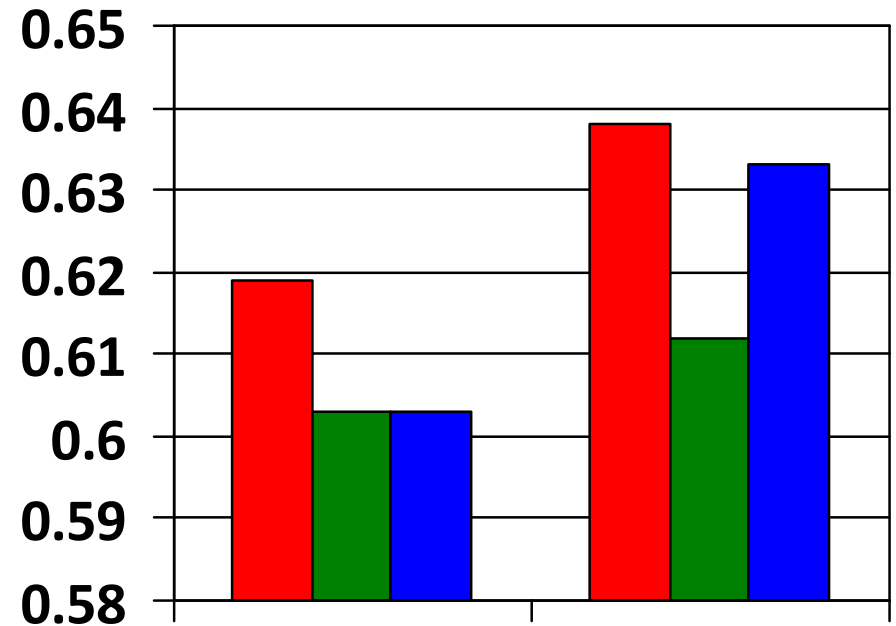
x



y

```
(((((.....))))).(((.....
(((.....)).....)))).....))))).)
```

ROC Area



Single

Grouped

■ Gradient ■ Direct ■ MM

Discussion

- How to choose α , β in Gamma prior?
 - Sensitivity experiments
 - Simple choice reasonable
 - Further investigation required
- Simple assumptions sometimes wrong
- But competitive performance with Grid, Grad
- Suited for ‘Quick-and-dirty’ implementations

Thank you!