The Concave-Convex Procedure (CCCP)

A. L. Yuille and Anand Rangarajan *

Smith-Kettlewell Eye Research Institute,

2318 Fillmore Street,

San Francisco, CA 94115, USA.

Tel. (415) 345-2144. Fax. (415) 345-8455.

Email yuille@ski.org

July 22, 2002

* Prof. Anand Rangarajan. Dept. of CISE, Univ. of Florida Room 301, CSE Building Gainesville, FL 32611-6120 Phone: (352) 392 1507 Fax: (352) 392 1220 e-mail: anand@cise.ufl.edu

Submitted to Neural Computation on 17 April 2002.

Revised 21 July 2002.

Abstract

The Concave-Convex procedure (CCCP) is a way to construct discrete time iterative dynamical systems which are guaranteed to monotonically decrease global optimization/energy functions. This procedure can be applied to almost any optimization problem and many existing algorithms can be interpreted in terms of it. In particular, we prove that all EM algorithms and classes of Legendre minimization and variational

bounding algorithms can be re-expressed in terms of CCCP. We show that many existing neural network and mean field theory algorithms are also examples of CCCP. The Generalized Iterative Scaling (GIS) algorithm and Sinkhorn's algorithm can also be expressed as CCCP by changing variables. CCCP can be used both as a new way to understand, and prove convergence of, existing optimization algorithms and as a procedure for generating new algorithms.

1 Introduction

This paper describes a simple geometrical Concave-Convex procedure (CCCP) for constructing discrete time dynamical systems which are guaranteed to decrease almost any global optimization/energy function monotonically. Such discrete time systems have advantages over standard gradient descent techniques (Press, Flannery, Teukolsky and Vetterling 1986) because they do not require estimating a step size and empirically often converge rapidly.

We first illustrate CCCP by giving examples of neural network, mean field, and self-annealing (which relate to Bregman distances (Bregman 1967)) algorithms which can be re-expressed in this form. As we will show, the entropy terms arising in mean field algorithms makes it particularly easy to apply CCCP. CCCP has also been applied to develop an algorithm which minimizes the Bethe and Kikuchi free energies and whose empirical convergence is rapid (Yuille 2002).

Next we prove that many existing algorithms can be directly re-expressed in terms of CCCP. This includes expectation-maximization (EM) algorithms (Dempster, Laird and Rubin 1977), minimization algorithms based on Legendre transforms (Rangarajan, Yuille and Mjolsness 1999). CCCP can be viewed as a special case of variational bounding (Rustagi 1976, Jordan, Ghahramani, Jaakkola and Saul 1999) and related techniques including lower bound maximization (Luttrell 1994), surrogate functions and majorization (Lange, Hunter

and Yang 2000). CCCP gives a novel geometric perspective on these algorithms and yields new convergence proofs.

Finally we reformulate other classic algorithms in terms of CCCP by changing variables. These include the Generalized Iterative Scaling (GIS) algorithm (Darroch and Ratcliff 1972) and Sinkhorn's algorithm for obtaining doubly stochastic matrices (Sinkhorn 1964). Sinkhorn's algorithm can be used to solve the linear assignment problem (Kosowsky and Yuille 1994) and CCCP variants of Sinkhorn can be used to solve additional constraint problems.

We introduce CCCP in section (2) and prove that it converges. Section (3) illustrates CCCP with examples from neural networks, mean field theory, self-annealing, and EM. In section (4.2) we prove the relationships between CCCP and the EM algorithm, Legendre transforms, and variational bounding. Section (5) shows that other algorithms such as GIS and Sinkhorn can be expressed in CCCP by a change of variables.

2 The Concave-Convex Procedure (CCCP)

This section introduces the main results of CCCP and summarizes them in three Theorems: (i) Theorem 1 states the general conditions under which CCCP can be applied, (ii) Theorem 2 defines CCCP and proves its convergence, and (iii) Theorem 3 describes an inner loop that may be necessary for some CCCP algorithms.

Theorem 1 shows that any function, subject to weak conditions, can be expressed as the sum of a convex and concave part (this decomposition is not unique). This will imply that CCCP can be applied to almost any optimization problem.

Theorem 1. *Let $E(\vec{x})$ be an energy function with bounded Hessian $\partial^2 E(\vec{x})/\partial\vec{x}\partial\vec{x}$. Then we can always decompose it into the sum of a convex function and a concave function.*

Proof. *Select any convex function $F(\vec{x})$ with positive definite Hessian with eigenvalues bounded below by $\epsilon > 0$. Then there exists a positive constant λ such that the Hessian of $E(\vec{x}) + \lambda F(\vec{x})$ is positive definite and hence $E(\vec{x}) + \lambda F(\vec{x})$ is convex. Hence we can express $E(\vec{x})$ as the sum of a convex part, $E(\vec{x}) + \lambda F(\vec{x})$, and a concave part $-\lambda F(\vec{x})$.*

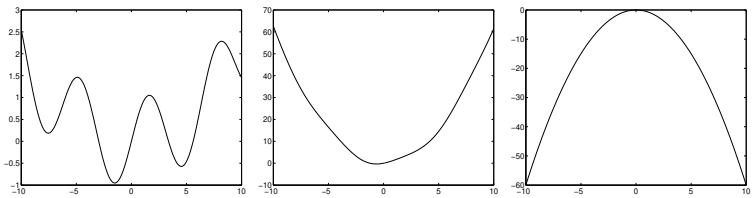

Figure 1: Decomposing a function into convex and concave parts. The original function (Left Panel) can be expressed as the sum of a convex function (Centre Panel) and a concave function (Right Panel).

Theorem 2 defines the CCCP procedure and proves that it converges to a minimum or a saddle point of the energy function. (After completing this work we found that a version of Theorem 2 appeared in an unpublished technical report by D. Geman (1984)).

Theorem 2. *Consider an energy function $E(\vec{x})$ (bounded below) of form $E(\vec{x}) = E_{vex}(\vec{x}) + E_{cave}(\vec{x})$ where $E_{vex}(\vec{x}), E_{cave}(\vec{x})$ are convex and concave functions of $\vec{x}$ respectively. Then the discrete iterative CCCP algorithm $\vec{x}^t \mapsto \vec{x}^{t+1}$ given by:*

$$\vec{\nabla} E_{vex}(\vec{x}^{t+1}) = -\vec{\nabla} E_{cave}(\vec{x}^t), \tag{1}$$

is guaranteed to monotonically decrease the energy $E(\vec{x})$ as a function of time and hence to converge to a minimum or saddle point of $E(\vec{x})$ (or even a local maxima if it starts at one). Moreover,

$$E(\vec{x}_{t+1}) = E(\vec{x}_t) - \frac{1}{2}(\vec{x}_{t+1} - \vec{x}_t)^T \{\vec{\nabla}\vec{\nabla} E_{vex}(\vec{x}^*) - \vec{\nabla}\vec{\nabla} E_{cave}(\vec{x}^{**})\}(\vec{x}_{t+1} - \vec{x}_t), \tag{2}$$

for some $\vec{x}^*$ and $\vec{x}^{**}$, where $\vec{\nabla}\vec{\nabla} E(.)$ is the Hessian of $E(.)$.

Proof. *The convexity and concavity of* $E_{vex}(.)$ *and* $E_{cave}(.)$ *means that* $E_{vex}(\vec{x}_2) \geq E_{vex}(\vec{x}_1) + (\vec{x}_2 - \vec{x}_1) \cdot \vec{\nabla} E_{vex}(\vec{x}_1)$ *and* $E_{cave}(\vec{x}_4) \leq E_{cave}(\vec{x}_3) + (\vec{x}_4 - \vec{x}_3) \cdot \vec{\nabla} E_{cave}(\vec{x}_3)$, *for all* $\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4$. *Now set* $\vec{x}_1 = \vec{x}^{t+1}, \vec{x}_2 = \vec{x}^t, \vec{x}_3 = \vec{x}^t, \vec{x}_4 = \vec{x}^{t+1}$. *Using the algorithm definition (i.e.* $\vec{\nabla} E_{vex}(\vec{x}^{t+1}) = -\vec{\nabla} E_{cave}(\vec{x}^t)$*) we find that* $E_{vex}(\vec{x}^{t+1}) + E_{cave}(\vec{x}^{t+1}) \leq E_{vex}(\vec{x}^t) + E_{cave}(\vec{x}^t)$, *which proves the first claim. The second claim follows by computing the second order terms of the Taylor series expansion and applying Rolle's theorem.*

We can get a graphical illustration of this algorithm by the reformulation shown in figure (2). Think of decomposing the energy function $E(\vec{x})$ into $E_1(\vec{x}) - E_2(\vec{x})$ where both $E_1(\vec{x})$ and $E_2(\vec{x})$ are convex. (This is equivalent to decomposing $E(\vec{x})$ into a a convex term $E_1(\vec{x})$ *plus* a concave term $-E_2(\vec{x})$). The algorithm proceeds by matching points on the two terms which have the same tangents. For an input $\vec{x}_0$ we calculate the gradient $\vec{\nabla} E_2(\vec{x}_0)$ and find the point $\vec{x}_1$ such that $\vec{\nabla} E_1(\vec{x}_1) = \vec{\nabla} E_2(\vec{x}_0)$. We next determine the point $\vec{x}_2$ such that $\vec{\nabla} E_1(\vec{x}_2) = \vec{\nabla} E_2(\vec{x}_1)$, and repeat.

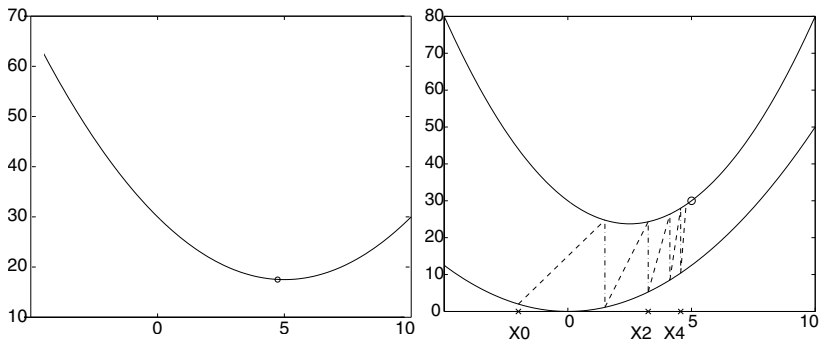

Figure 2: A CCCP algorithm illustrated for Convex minus Convex. We want to minimize the function in the Left Panel. We decompose it (Right Panel) into a convex part (top curve) minus a convex term (bottom curve). The algorithm iterates by matching points on the two curves which have the same tangent vectors, see text for more details. The algorithm rapidly converges to the solution at $x = 5.0$.

The second statement of Theorem 2 can be used to analyze the convergence rates of the

algorithm by placing lower bounds on the (positive semi-definite) matrix $\{\vec{\nabla}\vec{\nabla}E_{vex}(\vec{x}^*) - \vec{\nabla}\vec{\nabla}E_{cave}(\vec{x}^{**})\}$. Moreover, if we can bound this by a matrix $\mathbf{B}$ then we obtain $E(\vec{x}_{t+1}) - E(\vec{x}_t) \leq -(1/2)\{\vec{\nabla}E_{vex}^{-1}(-\vec{\nabla}E_{cave}(\vec{x}_t)) - \vec{x}_t\}^T\mathbf{B}\{\vec{\nabla}E_{vex}^{-1}(-\vec{\nabla}E_{cave}(\vec{x}_t)) - \vec{x}_t\} \leq 0$, where $\vec{\nabla}E_{vex}(\vec{x})^{-1}$ is the inverse of $\vec{\nabla}E_{vex}(\vec{x})$. We can therefore think of $(1/2)\{\vec{\nabla}E_{vex}^{-1}(-\vec{\nabla}E_{cave}(\vec{x}_t)) - \vec{x}_t\}^T\mathbf{B}\{\vec{\nabla}E_{vex}^{-1}(-\vec{\nabla}E_{cave}(\vec{x}_t)) - \vec{x}_t\}$ as an auxiliary function (Della Pietra, Della Pietra, and Lafferty 1997).

We can extend Theorem 2 to allow for linear constraints on the variables $\vec{x}$, for example $\sum_i c_i^\mu x_i = \alpha^\mu$ where the $\{c_i^\mu\}, \{\alpha^\mu\}$ are constants. This follows directly because properties such as convexity and concavity are preserved when linear constraints are imposed. We can change to new coordinates defined on the hyperplane defined by the linear constraints. Then we apply Theorem 1 in this coordinate system.

Observe that Theorem 2 defines the update as an *implicit* function of $\vec{x}^{t+1}$. In many cases, as we will show in section (3), it is possible to solve for $\vec{x}^{t+1}$ analytically. In other cases we may need an algorithm, or *inner loop*, to determine $\vec{x}^{t+1}$ from $\vec{\nabla}E_{vex}(\vec{x}^{t+1})$. In these cases we will need the following theorem where we re-express CCCP in terms of minimizing a time sequence of *convex update energy functions* $E_{t+1}(\vec{x}^{t+1})$ to obtain the updates $\vec{x}^{t+1}$ (i.e. at the t^{th} iteration of CCCP we need to minimize the energy $E_{t+1}(\vec{x}^{t+1})$).

Theorem 3. *Let $E(\vec{x}) = E_{vex}(\vec{x}) + E_{cave}(\vec{x})$ where $\vec{x}$ is required to satisfy the linear constraints $\sum_i c_i^\mu x_i = \alpha^\mu$, where the $\{c_i^\mu\}, \{\alpha^\mu\}$ are constants. Then the update rule for $\vec{x}^{t+1}$ can be formulated as setting $\vec{x}^{t+1} = \arg\min_{\vec{x}} E_{t+1}(\vec{x})$ for a time sequence of convex update energy functions $E_{t+1}(\vec{x})$ defined by:*

$$E_{t+1}(\vec{x}) = E_{vex}(\vec{x}) + \sum_i x_i \frac{\partial E_{cave}}{\partial x_i}(\vec{x}^t) + \sum_\mu \lambda_\mu \{\sum_i c_{i\mu} x_i - \alpha_\mu\}, \tag{3}$$

where the lagrange parameters $\{\lambda_\mu\}$ impose linear constraints.

Proof. *Direct calculation.*

The convexity of $E_{t+1}(\vec{x})$ implies that there is a unique minimum $\vec{x}^{t+1} = \arg\min_{\vec{x}} E_{t+1}(\vec{x})$. This means that if an inner loop is needed to calculate $\vec{x}^{t+1}$ then we can use standard

techniques such as conjugate gradient descent.

An important special case is when $E_{vex}(\vec{x}) = \sum_i x_i \log x_i$. This case occurs frequently in our examples, see section (3). We will show later in section (5) that $E_t(\vec{x})$ can be minimized by a CCCP algorithm.

3 Examples of CCCP

This section illustrates CCCP by examples from neural networks, mean field algorithms, self-annealing (which relate to Bregman distances (Bregman 1967)), EM and mixture models. These algorithms can be applied to a range of problems including clustering, combinatorial optimization, and learning.

Our first example is a neural net or mean field Potts model. These have been used for content addressable memories (Waugh and Westervelt 1993, Elfadel 1995). They have also been applied to clustering for unsupervised texture segmentation (Hofmann and Buhmann 1997). An original motivation for them was based on a convexity principle (Marcus and Westervelt 1989). We now show that algorithms for these models can be derived directly using CCCP.

Example 1. *Discrete Time Dynamical Systems for the Mean Field Potts Model. These attempt to minimize discrete energy functions of form* $E[V] = (1/2)\sum_{i,j,a,b} C_{ijab} V_{ia} V_{jb} + \sum_{ia} \theta_{ia} V_{ia}$, *where the* $\{V_{ia}\}$ *take discrete values* $\{0, 1\}$ *with linear constraints* $\sum_i V_{ia} = 1, \ \forall a$.

Discussion. *Mean field algorithms minimize a continuous effective energy* $E_{eff}[S; T]$ *to obtain a minimum of the discrete energy* $E[V]$ *in the limit as* $T \mapsto 0$. *The* $\{S_{ia}\}$ *are continuous variables in the range* $[0, 1]$ *and correspond to (approximate) estimates of the mean states of the* $\{V_{ia}\}$ *with respect to the distribution* $P[V] = e^{-E[V]/T}/Z$, *where* T *is a temperature parameter and* Z *is a normalization constant. As described in (Yuille and Kosowsky 1994),*

to ensure that the minima of $E[V]$ *and* $E_{eff}[S;T]$ *all coincide (as* $T \mapsto 0$*) it is sufficient that* C_{ijab} *be negative definite. Moreover, this can be attained by adding a term* $-K\sum_{ia} V_{ia}^2$ *to* $E[V]$ *(for sufficiently large* K*) without altering the structure of the minima of* $E[V]$*. Hence, without loss of generality we can consider* $(1/2)\sum_{i,j,a,b} C_{ijab}V_{ia}V_{jb}$ *to be a concave function.*

We impose the linear constraints by adding a Lagrange multiplier term $\sum_a p_a\{\sum_i V_{ia} - 1\}$ *to the energy where the* $\{p_a\}$ *are the Lagrange multipliers. The effective energy is given by:*

$$E_{eff}[S] = (1/2)\sum_{i,j,a,b} C_{ijab}S_{ia}S_{jb} + \sum_{ia}\theta_{ia}S_{ia} + T\sum_{ia} S_{ia}\log S_{ia} + \sum_a p_a\{\sum_i S_{ia} - 1\}. \quad (4)$$

We decompose $E_{eff}[S]$ *into a convex part* $E_{vex} = T\sum_{ia} S_{ia}\log S_{ia} + \sum_a p_a\{\sum_i S_{ia} - 1\}$ *and a concave part* $E_{cave}[S] = (1/2)\sum_{i,j,a,b} C_{ijab}S_{ia}S_{jb} + \sum_{ia}\theta_{ia}S_{ia}$*. Taking derivatives yields:* $\frac{\partial}{\partial S_{ia}}E_{vex}[S] = T\log S_{ia} + p_a$ *and* $\frac{\partial}{\partial S_{ia}}E_{cave}[S] = \sum_{j,b} C_{ijab}S_{jb} + \theta_{ia}$*. Applying CCCP by setting* $\frac{\partial E_{vex}}{\partial S_{ia}}(S^{t+1}) = -\frac{\partial E_{cave}}{\partial S_{ia}}(S^t)$ *gives* $T\{1 + \log S_{ia}^{t+1}\} + p_a = -\sum_{j,b} C_{ijab}S_{jb}^t - \theta_{ia}$*. We solve for the Lagrange multipliers* $\{p_a\}$ *by imposing the constraints* $\sum_i S_{ia}^{t+1} = 1,\ \forall\ a$*. This gives a discrete update rule:*

$$S_{ia}^{t+1} = \frac{e^{(-1/T)\{2\sum_{j,b} C_{ijab}S_{jb}^t + \theta_{ia}\}}}{\sum_c e^{(-1/T)\{2\sum_{j,b} C_{ijcb}S_{jb}^t + \theta_{ic}\}}}. \quad (5)$$

The next example concerns mean field methods to model combinatorial optimization problems such as the quadratic assignment problem (Rangarajan, Gold and Mjolsness 1996, Rangarajan, Yuille and Mjolsness 1999) or the Travelling Salesman Problem. It uses the same quadratic energy function as the last example but adds extra linear constraints. These additional constraints prevent us from expressing the update rule analytically and require an inner loop to implement Theorem 3.

Example 2. *Mean Field Algorithms to minimize discrete energy functions of form* $E[V] = \sum_{i,j,a,b} C_{ijab}V_{ia}V_{jb} + \sum_{ia}\theta_{ia}V_{ia}$ *with linear constraints* $\sum_i V_{ia} = 1,\ \forall a$ *and* $\sum_a V_{ia} = 1,\ \forall i$*.*

Discussion. *This differs from the previous example because we need to add an additional constraint term* $\sum_i q_i(\sum_a S_{ia} - 1)$ *to the effective energy* $E_{eff}[S]$ *in equation (4) where* $\{q_i\}$ *are Lagrange multipliers. This constraint term is also added to the convex part of the energy* $E_{vex}[S]$ *and we apply CCCP. Unlike the previous example, it is no longer possible to express* S^{t+1} *as an analytic function of* S^t*. Instead we resort to Theorem 3. Solving for* S^{t+1} *is equivalent to minimizing the convex cost function:*

$$E_{t+1}[S^{t+1}; p, q] = T \sum_{ia} S_{ia}^{t+1} \log S_{ia}^{t+1} + \sum_a p_a \{\sum_i S_{ia}^{t+1} - 1\} + \sum_i q_i \{\sum_a S_{ia}^{t+1} - 1\} + \sum_{ia} S_{ia}^{t+1} \frac{\partial E_{cave}}{S_{ia}}(S_{ia}^t). \quad (6)$$

It can be shown that minimizing $E_{t+1}[S^{t+1}; p, q]$ *can also be done by CCCP, see section (5.3). Therefore each step of CCCP for this example requires an inner loop which in turn can be solved by a CCCP algorithm.*

Our next example is self-annealing (Rangarajan 2000). This algorithm can be applied to the effective energies of the last two examples provided we remove the "entropy term" $\sum_{ia} S_{ia} \log S_{ia}$. Hence self-annealing can be applied to the same combinatorial optimization and clustering problems. It can also be applied to the relaxation labelling problems studied in computer vision and indeed the classic relaxation algorithm (Rosenfeld, Hummel and Zucker 2000) can be obtained as an approximation to self-annealing by performing a Taylor series approximation (Rangarajan 2000). It also relates to linear prediction (Kivinen and Warmuth 1997).

Self-annealing acts as if it has a temperature parameter which it continuously decreases or, equivalently, as if it has a barrier function whose strength is reduced automatically as the algorithm proceeds (Rangarajan 2000). This relates to Bregman distances (Bregman 1967) and, indeed, the original derivation of self-annealing involved adding a Bregman distance to the energy function followed by taking Legendre transforms, see section (4.2). As we now

show, however, self-annealing can be derived directly from CCCP.

Example 3. *Self-Annealing for quadratic energy functions. We use the effective energy of example 1, see equation (4), but remove the entropy term* $T \sum_{ia} S_{ia} \log S_{ia}$. *We first apply both sets of linear constraints on the* $\{S_{ia}\}$ *(as in example 2). Next we apply only one set of constraints (as in example 1).*

Discussion. *Decompose the energy function into convex and concave parts by adding and subtracting a term* $\gamma \sum_{i,a} S_{ia} \log S_{ia}$, *where* γ *is a constant. This yields:*

$$E_{vex}[S] = \gamma \sum_{ia} S_{ia} \log S_{ia} + \sum_{a} p_a (\sum_{i} S_{ia} - 1) + \sum_{i} q_i (\sum_{a} S_{ia} - 1),$$
$$E_{cave}[S] = \frac{1}{2} \sum_{i,j,a,b} C_{ijab} S_{ia} S_{jb} + \sum_{i,a} \theta_{ia} S_{ia} - \gamma \sum_{ia} S_{ia} \log S_{ia}. \tag{7}$$

Applying CCCP gives the self-annealing update equations:

$$S_{ia}^{t+1} = S_{ia}^{t} e^{(1/\gamma)\{-\sum_{jb} C_{ijab} S_{jb}^{t} - \theta_{ia} - p_a - q_i\}}, \tag{8}$$

where an inner loop is required to solve for the $\{p_a\}, \{q_i\}$ *to ensure that the constraints on* $\{S_{ia}^{t+1}\}$ *are satisfied. This inner loop is a small modification of the one required for example 2, see section (5.3).*

Removing the constraints $\sum_i q_i (\sum_a S_{ia} - 1)$ *gives us an update rule (compare example 1):*

$$S_{ia}^{t+1} = \frac{S_{ia}^{t} e^{(-1/\gamma)\{\sum_{jb} C_{ijab} S_{jb}^{t} + \theta_{ia}\}}}{\sum_c S_{ic}^{t} e^{(-1/\gamma)\{\sum_{jb} C_{ijbc} S_{bj}^{t} + \theta_{ic}\}}}, \tag{9}$$

which, by expanding the exponential by a Taylor series, gives the equations for relaxation labelling (Rosenfeld, Hummel and Zucker 1976) (see (Rangarajan 2000) for details).

Our final example is the elastic net (Durbin and Willshaw 1987, Durbin, Szeliski and Yuille 1989) in the formulation presented in (Yuille 1990). This is an example of constrained mixture models (Jordan and Jacobs 1994) and uses an EM algorithm (Dempster, Laird and Rubin 1977).

Example 4. *The elastic net (Durbin and Willshaw 1987) attempts to solve the Travelling Salesman Problem (TSP) by finding the shortest tour through a set of cities at positions* $\{\vec{x}_i\}$*. The net is represented by a set of nodes at positions* $\{\vec{y}_a\}$ *and the algorithm performs steepest descent on a cost function* $E[\vec{y}]$*. This corresponds to a probability distribution* $P(\vec{y}) = e^{-E[\vec{y}]}/Z$ *on the node positions which can be interpreted (Durbin, Szeliski and Yuille 1989) as a constrained mixture model (Jordan and Jacobs 1994). The elastic net can be reformulated (Yuille 1990) as minimizing an effective energy* $E_{eff}[S, \vec{y}]$ *where the variables* $\{S_{ia}\}$ *determine soft correspondence between the cities and the nodes of the net. Minimizing* $E_{eff}[S, \vec{y}]$ *with respect to* S *and* $\vec{y}$ *alternatively can be reformulated as a CCCP algorithm. Moreover, this alternating algorithm can also be re-expressed as an EM algorithm for performing MAP estimation of the node variables* $\{\vec{y}_a\}$ *from* $P(\vec{y})$*, see section (4.1).*

Discussion. *The elastic net can be formulated as minimizing an effective energy (Yuille 1990):*

$$E_{eff}[S, \vec{y}] = \sum_{ia} S_{ia}(\vec{x}_i - \vec{y}_a)^2 + \gamma \sum_{a,b} \vec{y}_a^T A_{ab} \vec{y}_b + T \sum_{i,a} S_{ia} \log S_{ia} + \sum_i \lambda_i (\sum_a S_{ia} - 1), \quad (10)$$

where the $\{A_{ab}\}$ *are components of a positive definite matrix representing a spring energy and* $\{\lambda_a\}$ *are Lagrange multipliers which impose the constraints* $\sum_a S_{ia} = 1, \ \forall \ i$*. By setting* $E[\vec{y}] = E_{eff}[S^*(\vec{y}), \vec{y}]$ *where* $S^*(\vec{y}) = \arg\min_S E_{eff}[S, \vec{y}]$*, we obtain the original elastic net cost function* $E[\vec{y}] = -T \sum_i \log \sum_a e^{-|\vec{x}_i - \vec{y}_a|^2/T} + \gamma \sum_{a,b} \vec{y}_a^T A_{ab} \vec{y}_b$ *(Durbin and Willshaw 1987).* $P[\vec{y}] = e^{-E[\vec{y}]}/Z$ *can be interpreted (Durbin, Szeliski and Yuille 1989) as a constrained mixture model (Jordan and Jacobs 1994).*

The effective energy $E_{eff}[S, \vec{y}]$ *can decreased by minimizing it with respect to* $\{S_{ia}\}$ *and* $\{\vec{y}_a\}$ *alternatively. This gives update rules:*

$$S_{ia}^{t+1} = \frac{e^{-|\vec{x}_i - \vec{y}_a^t|^2/T}}{\sum_j e^{-|\vec{x}_j - \vec{y}_a^t|^2/T}}, \quad (11)$$

$$\sum_i S_{ia}^{t+1} (\vec{y}_a^{t+1} - \vec{x}_i) + \sum_b A_{ab} \vec{y}_b^{t+1} = 0 \ \forall \ a, \quad (12)$$

where $\{\vec{y}_a^{t+1}\}$ *can be computed from the* $\{S_{ia}^{t+1}\}$ *by solving the linear equations.*

To interpret equations (11,12) as CCCP, we define a new energy function $E[S] = E_{eff}[S, \vec{y}^*(S)]$ *where* $\vec{y}^*(S) = \arg\min_{\vec{y}} E_{eff}[S, \vec{y}]$ *(which can be obtained by solving the linear equation (12) for* $\{\vec{y}_a\}$*). We decompose* $E[S] = E_{vex}[S] + E_{cave}[S]$ *where:*

$$E_{vex}[S] = T \sum_{ia} S_{ia} \log S_{ia} + \sum_i \lambda_i (\sum_a S_{ia} - 1),$$
$$E_{cave}[S] = \sum_{ia} S_{ia} |\vec{x}_i - \vec{y}_a^*(S)|^2 + \gamma \sum_{ab} \vec{y}_a^*(S) \cdot \vec{y}_b^*(S) A_{ab}. \tag{13}$$

It is clear that $E_{vex}[S]$ *is a convex function of* S*. It can be verified algebraically that* $E_{cave}[S]$ *is a concave function of* S *and that its first derivative is* $\frac{\partial}{\partial S_{ia}} E_{cave}[S] = |\vec{x}_i - \vec{y}_a^*(S)|^2$ *(using the definition of* $\vec{y}^*(S)$ *to remove additional terms). Applying CCCP to* $E[S] = E_{vex}[S] + E_{cave}[S]$ *gives the update rule:*

$$S_{ia}^{t+1} = \frac{e^{-|\vec{x}_i - \vec{y}_a^*(S^t)|^2}}{\sum_b e^{-|\vec{x}_i - \vec{y}_b^*(S^t)|^2}}, \quad \vec{y}^*(S^t) = \arg\min_{\vec{y}} E_{eff}[S^t, \vec{y}], \tag{14}$$

which is equivalent to the alternating algorithm described above, see equations (11,12).

More understanding of this particular CCCP algorithm is given in the next section where we show that is a special case of a general result for EM algorithms.

4 EM, Legendre Transforms, and Variational Bounding

This section proves that two standard algorithms can be expressed in terms of CCCP: (i) all EM algorithms, and (ii) a class of algorithms using Legendre transforms. In addition, we show that CCCP can be obtained as a special case of variational bounding and equivalent methods known as lower bound maximization and surrogate functions.

4.1 The EM algorithm and CCCP

The EM algorithm (Dempster, Laird and Rubin 1977) seeks to estimate a variable $\vec{y}^* = \arg\max_{\vec{y}} \log \sum_{\{V\}} P(\vec{y}, V)$, where $\{\vec{y}\}, \{V\}$ are variables that depend on the specific problem formulation (we will soon illustrate them for the elastic net). The distribution $P(\vec{y}, V)$ is usually conditioned on data which, for simplicity, we will not make explicit.

It was shown in (Hathaway 1986, Neal and Hinton 1998) that EM is equivalent to *minimizing* the following effective energy with respect to the variables $\vec{y}$ and $\hat{P}(V)$:

$$E_{em}[\vec{y}, \hat{P}] = -\sum_V \hat{P}(V) \log P(\vec{y}, V) + \sum_V \hat{P}(V) \log \hat{P}(V) + \lambda\{\sum \hat{P}(V) - 1\}, \tag{15}$$

where λ is a lagrange multiplier.

The EM algorithm proceeds by minimizing $E_{em}[\vec{y}, \hat{P}]$ with respect to $\hat{P}(V)$ and $\vec{y}$ alternatively:

$$\hat{P}^{t+1}(V) = \frac{P(\vec{y}^t, V)}{\sum_{\hat{V}} P(\vec{y}^t, \hat{V})}, \quad \vec{y}^{t+1} = \arg\min_{\vec{y}} -\sum_V \hat{P}^{t+1}(V) \log P(\vec{y}, V). \tag{16}$$

These update rules are guaranteed to lower $E_{em}[\vec{y}, \hat{P}$ and give convergence to a saddle point or a local minimum (Dempster, Laird and Rubin 1977, Hathaway 1986, Neal and Hinton 1998).

For example, this formulation of the EM algorithm enables us to rederive the effective energy for the elastic net and show that the alternating algorithm is EM. We let the $\{\vec{y}_a\}$ correspond to the positions of the nodes of the net and the $\{V_{ia}\}$ be binary variables indicating the correspondences between cities and nodes (related to the $\{S_{ia}\}$ in example 4). $P(\vec{y}, V) = e^{-E[\vec{y},V]/T}/Z$ where $E[\vec{y}, V] = \sum_{ia} V_{ia}|\vec{x}_i - \vec{y}_a|^2 + \gamma \sum_{ab} \vec{y}_a \cdot \vec{y}_b A_{ab}$ with constraint that $\sum_a V_{ia} = 1, \ \forall \ i$. We define $S_{ia} = \hat{P}(V_{ia} = 1), \ \forall \ i, a$. Then $E_{em}[\vec{y}, S]$ is equal to the effective energy $E_{eff}[\vec{y}, S]$ in the elastic net example, see equation (10). The update rules for EM, see equation (16), are equivalent to the alternating algorithm to minimize the effective energy,

see equations (11,12).

We now show that all EM algorithms are CCCP. This requires two intermediate results which we state as lemmas.

Lemma 1. *Minimizing $E_{em}[\vec{y}, \hat{P}]$ is equivalent to minimizing the function $E[\hat{P}] = E_{vex}[\hat{P}] + E_{cave}[\hat{P}]$ where $E_{vex}[\hat{P}] = \sum_V \hat{P}(V) \log \hat{P}(V) + \lambda\{\sum \hat{P}(V) - 1\}$ is a convex function and $E_{cave}[\hat{P}] = -\sum_V \hat{P}(V) \log P(\vec{y}^*(\hat{P}), V)$ is a concave function, where we define $\vec{y}^*(\hat{P}) = \arg\min_{\vec{y}} - \sum_V \hat{P}(V) \log P(\vec{y}, V)$.*

Proof. *Set $E[\hat{P}] = E_{em}[\vec{y}^*(\hat{P}), \hat{P}]$, where $\vec{y}^*(\hat{P}) = \arg\min_{\vec{y}} - \sum_V \hat{P}(V) \log P(\vec{y}, V)$. It is straightforward to decompose $E[\hat{P}]$ as $E_{vex}[\hat{P}] + E_{cave}[\hat{P}]$ and verify that $E_{vex}[\hat{P}]$ is a convex function. To determine that $E_{cave}[\hat{P}]$ is concave requires showing that its Hessian is negative semi-definite. This is performed in lemma 2.*

Lemma 2. *$E_{cave}[\hat{P}]$ is a concave function and $\frac{\partial}{\partial \hat{P}(V)} E_{cave} = -\log P(\vec{y}^*(\hat{P}), V)$, where $\vec{y}^*(\hat{P}) = \arg\min_{\vec{y}} - \sum_V \hat{P}(V) \log P(\vec{y}, V)$.*

Proof. *We first derive consequences of the definition of $\vec{y}^*$ which will be required when computing the Hessian of E_{cave}. The definition implies:*

$$\sum_V \hat{P}(V) \frac{\partial}{\partial \vec{y}_\mu} \log P(\vec{y}^*, V) = 0 \ \forall \mu \tag{17}$$

$$\frac{\partial}{\partial \vec{y}_\mu} \log P(\vec{y}^*, \tilde{V}) + \sum_V \hat{P}(V) \sum_\nu \frac{\partial \vec{y}^*_\nu}{\partial \hat{P}(\tilde{V})} \frac{\partial^2}{\partial \vec{y}_\mu \partial \vec{y}_\nu} \log P(\vec{y}^*, V) = 0, \ \forall \ \mu., \tag{18}$$

where the first equation is an identity which is valid for all $\hat{P}$ and the second equation follows by differentiating the first equation with respect to $\hat{P}$. Moreover, since $\vec{y}^*$ is a minimum of $-\sum_V \hat{P}(V) \log P(\vec{y}, V)$ we also know that the matrix $\sum_V \hat{P}(V) \frac{\partial^2}{\partial y_\mu \partial y_\nu} \log P(\vec{y}^*, V)$ is negative definite. (We use the convention that $\frac{\partial}{\partial \vec{y}_\mu} \log P(\vec{y}^*, V)$ denotes the derivative of the function $\log P(\vec{y}, V)$ with respect to $\vec{y}_\mu$ evaluated at $\vec{y} = \vec{y}^*$).

We now calculate the derivatives of $E_{conv}[\hat{P}]$ *with respect to* $\hat{P}$. *We obtain:*

$$\frac{\partial}{\partial \hat{P}(\tilde{V})} E_{cave} = -\log P(\vec{y}^*(\hat{P}), \tilde{V}) - \sum_V \hat{P}(V) \sum_{mu} \frac{\partial \vec{y}^*_\mu}{\partial \hat{P}(\tilde{V})} \frac{\partial \log P(\vec{y}^*, V)}{\partial f_\mu} = -\log P(\vec{y}^*, V), \tag{19}$$

where we have used the definition of $\vec{y}^*$, *see equation (17), to eliminate the second term on the right hand side. This proves the first statement of the theorem.*

To prove the concavity of E_{cave}, *we compute its Hessian:*

$$\frac{\partial^2}{\partial \hat{P}(V) \partial \hat{P}(\tilde{V})} E_{cave} = -\sum_\nu \frac{\partial \vec{y}^*_\nu}{\partial \hat{P}(\tilde{V})} \frac{\partial}{\partial \vec{y}_\nu} \log P(\vec{y}^*, V). \tag{20}$$

By using the definition of $\vec{y}^*(\hat{P})$, *see equation (18), we can re-express the Hessian as:*

$$\frac{\partial^2}{\partial \hat{P}(V) \partial \hat{P}(\tilde{V})} E_{cave} = \sum_{V,\mu,\nu} \frac{\partial \vec{y}^*_\nu}{\partial \hat{P}(\tilde{V})} \frac{\partial \vec{y}^*_\mu}{\partial \hat{P}(V)} \frac{\partial^2 \log P(\vec{y}^*, V)}{\partial \vec{y}_\mu \partial \vec{y}_\nu}. \tag{21}$$

It follows that E_{cave} *has a negative definite Hessian and hence* E_{cave} *is concave, recalling that* $-\sum_V \hat{P}(V) \frac{\partial^2}{\partial y_\mu \partial y_\nu} \log P(\vec{y}^*, V)$ *is negative definite.*

Theorem 4. *The EM algorithm for* $P(\vec{y}, V)$ *can be expressed as a CCCP algorithm in* $\hat{P}(V)$ *with* $E_{vex}[\hat{P}] = \sum_V \hat{P}(V) \log \hat{P}(V) + \lambda\{\sum \hat{P}(V) - 1\}$ *and* $E_{cave}[\hat{P}] = -\sum_V \hat{P}(V) \log P(\vec{y}^*(\hat{P}), V)$, *where* $\vec{y}^*(\hat{P}) = \arg\min_{\vec{y}} -\sum_V \hat{P}(V) \log P(\vec{y}, V)$. *After convergence to* $\hat{P}^*(V)$, *the solution is calculated to be* $\vec{y}^{**} = \arg\min_{\vec{y}} -\sum_V \hat{P}^*(V) \log P(\vec{y}, V)$.

Proof. *The update rule for* $\hat{P}$ *determined by CCCP is precisely that specified by the EM algorithm. Therefore we can run the CCCP algorithm until it has converged to* $\hat{P}^*(.)$ *and then calculate the solution* $\vec{y}^{**} = \arg\min_{\vec{y}} -\sum_V \hat{P}^*(V) \log P(\vec{y}, V)$.

Finally, we observe that D. Geman's technical report (Geman 1984) gives an alternative way of relating EM to CCCP for a special class of probability distributions. He assumes that $P(\vec{y}, V)$ is of form $e^{\vec{y} \cdot \vec{\phi}(V)}/Z$ for some functions $\vec{\Phi}(.)$. He then proves convergence of the EM algorithm to estimate $\vec{y}$ by exploiting his version of Theorem 2. Interestingly, he works with convex and concave functions of $\vec{y}$ while our results are expressed in terms of convex and concave functions of $\hat{P}$.

4.2 Legendre Transformations

The Legendre transform can be used to reformulate optimization problems by introducing auxiliary variables (Mjolsness and Garrett 1990). The idea is that some of the formulations may be more effective and computationally cheaper than others.

We will concentrate on *Legendre minimization*, see (Rangarajan, Gold and Mjolsness 1996, Rangarajan, Yuille and Mjolsness 1999), instead of *Legendre min-max* emphasized in (Mjolsness and Garrett 1990). In the later, see (Mjolsness and Garrett 1990), the introduction of auxiliary variables converts the problem to a min-max problem where the goal is to find a saddle point. By contrast, in *Legendre minimization*, see (Rangarajan, Gold and Mjolsness 1996), the problem remains a minimization one (and so it becomes easier to analyze convergence).

In Theorem 5 we show that *Legendre minimization* algorithms are equivalent to CCCP provided we first decompose the energy into a convex plus a concave part. The CCCP viewpoint emphasizes the geometry of the approach and complements the algebraic manipulations given in (Rangarajan, Yuille and Mjolsness 1999). (Moreover, the results of this paper show the generality of CCCP while, by contrast, Legendre transform methods have been applied only on a case by case basis).

Definition 1. *Let $F(\vec{x})$ be a convex function. For each value $\vec{y}$ let $F^*(\vec{y}) = \min_{\vec{x}}\{F(\vec{x}) + \vec{y} \cdot \vec{x}.\}$. Then $F^*(\vec{y})$ is concave and is the Legendre transform of $F(\vec{x})$.*

Two properites can be derived from this definition (Strang 1986).

Property 1. $F(\vec{x}) = \max_{\vec{y}}\{F^*(\vec{y}) - \vec{y} \cdot \vec{x}\}$.

Property 2. *$F(.)$ and $F^*(.)$ are related by $\frac{\partial F^*}{\partial \vec{y}}(\vec{y}) = \{\frac{\partial F}{\partial \vec{x}}\}^{-1}(-\vec{y})$, $-\frac{\partial F}{\partial \vec{x}}(\vec{x}) = \{\frac{\partial F^*}{\partial \vec{y}}\}^{-1}(\vec{x})$. (By $\{\frac{\partial F^*}{\partial \vec{y}}\}^{-1}(\vec{x})$ we mean the value $\vec{y}$ such that $\frac{\partial F^*}{\partial \vec{y}}(\vec{y}) = \vec{x}$.)*

The Legendre minimization algorithms (Rangarajan, Gold and Mjolsness 1996, Rangarajan, Yuille and Mjolsness 1999) exploits Legendre transforms. The optimization function

$E_1(\vec{x})$ is expressed as $E_1(\vec{x}) = f(\vec{x}) + g(\vec{x})$ where $g(\vec{x})$ is required to be a convex function. This is equivalent to minimizing $E_2(\vec{x}, \vec{y}) = f(\vec{x}) + \vec{x} \cdot \vec{y} + \hat{g}(\vec{y})$, where $\hat{g}(.)$ is the inverse Legendre transform of $g(.)$. Legendre minimization consists of minimizing $E_2(\vec{x}, \vec{y})$ with respect to $\vec{x}$ and $\vec{y}$ alternatively.

Theorem 5. *Let $E_1(\vec{x}) = f(\vec{x}) + g(\vec{x})$ and $E_2(\vec{x}, \vec{y}) = f(\vec{x}) + \vec{x} \cdot \vec{y} + h(\vec{y})$, where $f(.), h(.)$ are convex functions and $g(.)$ is concave. Then applying CCCP to $E_1(\vec{x})$ is equivalent to minimizing $E_2(\vec{x}, \vec{y})$ with respect to $\vec{x}$ and $\vec{y}$ alternatively, where $g(.)$ is the Legendre transform of $h(.)$. This is equivalent to Legendre minimization.*

Proof. *We can write $E_1(\vec{x}) = f(\vec{x}) + \min_{\vec{y}}\{g^*(\vec{y}) + \vec{x} \cdot \vec{y}\}$ where $g^*(.)$ is the Legendre transform of $g(.)$ (identify $g(.)$ with $F^*(.)$ and $g^*(.)$ with $F(.)$ in Definition 1 and Property 1). Thus minimizing $E_1(\vec{x})$ with respect to $\vec{x}$ is equivalent to minimizing $\hat{E}_1(\vec{x}, \vec{y}) = f(\vec{x}) + \vec{x} \cdot \vec{y} + g^*(\vec{y})$ with respect to $\vec{x}$ and $\vec{y}$. (Alternatively, we can set $g^*(\vec{y}) = h(\vec{y})$ in the expression for $E_2(\vec{x}, \vec{y})$ and obtain a cost function $\hat{E}_2(\vec{x}) = f(\vec{x}) + g(\vec{x})$.) Alternatively minimization over $\vec{x}$ and $\vec{y}$ gives: (i) $\partial f / \partial \vec{x} = \vec{y}$ to determine $\vec{x}^{t+1}$ in terms of $\vec{y}^t$, and (ii) $\partial g^* / \partial \vec{y} = \vec{x}$ to determine $\vec{y}^t$ in terms of $\vec{x}^t$ which, by Property 2 of the Legendre transform is equivalent to setting $\vec{y} = -\partial g / \partial \vec{x}$. Combining these two stages gives CCCP:*

$$\frac{\partial f}{\partial \vec{x}}(\vec{x}^{t+1}) = -\frac{\partial g}{\partial \vec{x}}(\vec{x}^t).$$

4.3 Variational Bounding

In variational bounding, the original objective function to be minimized gets replaced by a new objective function which satisfies the following requirements, see (Rustagi 1976, Jordan, Ghahramani, Jaakkola and Saul 1999). Other equivalent techniques are known as surrogate functions and majorization (Lange, Hunter and Yang 2000) or as upper bound maximization

(Luttrell 1994). These techniques are more general than CCCP and it has been shown that algorithms like EM can be derived from them (Minka 1998, Lange, Hunter and Yang 2000).

Let $E(\vec{x}),\ \vec{x} \in \mathcal{R}^D$ be the original objective function that we seek to minimize. Assume that we are at a point $\vec{x}^{(n)}$ corresponding to the nth iteration. If we have a function $E_{bound}(\vec{x})$ which satisfies the following properties, see figure (3),

$$E(\vec{x}^{(n)}) = E_{bound}(\vec{x}^{(n)}),\ \text{and} \tag{22}$$

$$E(\vec{x}) \leq E_{bound}(\vec{x}) \tag{23}$$

then the next iterate $\vec{x}^{(n+1)}$ is chosen such that

$$E_{bound}(\vec{x}^{(n+1)}) \leq E(\vec{x}^{(n)}) \text{ whichimplies } E(\vec{x}^{(n+1)}) \leq E(\vec{x}^{(n)}). \tag{24}$$

Consequently, we can minimize $E_{bound}(\vec{x})$ instead of $E(\vec{x})$ after ensuring that $E(\vec{x}^{(n)}) = E_{bound}(\vec{x}^{(n)})$.

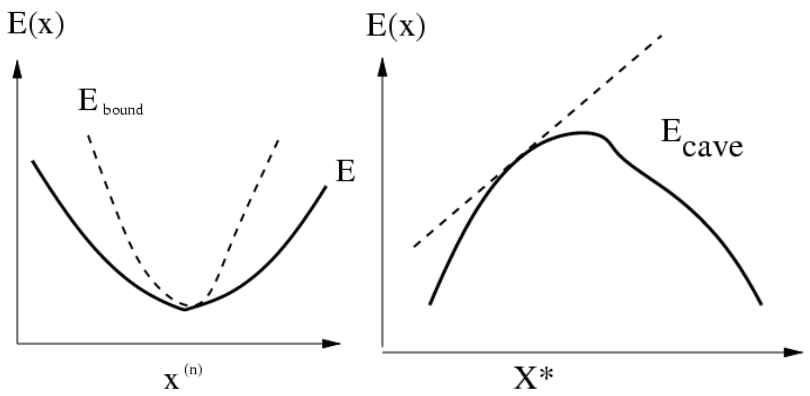

Figure 3: Variational bounding, see Left Panel, bounds a function $E(\vec{x})$ by a function $E_{bound}(\vec{x})$ such that $E(\vec{x}^{(n)}) = E_{bound}(\vec{x}^{(n)})$. We decompose $E(\vec{x})$ into convex and concave parts, $E_{vex}(\vec{x})$ and $E_{cave}(\vec{x})$ and bound $E_{cave}(\vec{x})$ by its tangent plane at $\vec{x}^*$, see Right Panel. We set $E_{bound}(\vec{x}) = E_{vex}(\vec{x}) + E_{cave}(\vec{x})$.

We now show that CCCP is equivalent to a class of variational bounding provided we first decompose the objective function $E(\vec{x})$ into a convex and a concave part before bounding

the concave part by its tangent plane, see figure (3).

Theorem 6. *Any CCCP algorithm to extremize $E(\vec{x})$ can be expressed as variational bounding by first decomposing $E(\vec{x})$ as a convex $E_{vex}(\vec{x})$ and a concave $E_{cave}(\vec{x})$ function and then at each iteration starting at $\vec{x}^t$ set $E^t_{bound}(\vec{x}) = E_{vex}(\vec{x}) + E_{cave}(\vec{x}^t) + (\vec{x} - \vec{x}^t) \cdot \frac{\partial E_{cave}(\vec{x}^t)}{\partial \vec{x}} \geq E_{cave}(\vec{x})$.*

Proof. *Since $E_{cave}(\vec{x})$ is concave, we have $E_{cave}(\vec{x}^*) + (\vec{x} - \vec{x}^*) \cdot \frac{\partial E_{cave}}{\partial \vec{x}} \geq E_{cave}(\vec{x})$ for all $\vec{x}$. Therefore $E^t_{bound}(\vec{x})$ satisfies the equations (22,23) for variational bounding. Minimizing $E^t_{bound}(\vec{x})$ with respect to $\vec{x}$ gives $\frac{\partial}{\partial \vec{x}} E_{vex}(\vec{x}^{t+1}) = -\frac{\partial E_{cave}(\vec{x}^t)}{\partial \vec{x}}$ which is the CCCP update rule.*

Note that the formulation of CCCP given earlier by Theorem 3, in terms of a sequence of convex update energy functions, is already in the variational bounding form.

5 CCCP by changes in variables

This section gives examples where the algorithms are not CCCP in the original variables. But they can be transformed into CCCP by changing coordinates. In this section, we first show that Generalized Iterative Scaling (GIS) (Darroch and Ratcliff 1972) and Sinkhorn's algorithm (Sinkhorn 1964) can both be formulated as CCCP. Then we obtain CCCP generalizations of Sinkhorn's algorithm which can minimize many of the inner-loop convex update energy functions defined in Theorem 3.

5.1 Generalized Iterative Scaling

This section shows that the Generalized Iterative Scaling (GIS) algorithm (Darroch and Ratcliff 1972) for estimating parameters of probability distributions can also be expressed as CCCP. This gives a simple converge proof for the algorithm.

The parameter estimation problem is to determine the parameters $\vec{\lambda}$ of a distribution

$P(\vec{x} : \vec{\lambda}) = e^{\vec{\lambda}\cdot\vec{\phi}(\vec{x})}/Z[\vec{\lambda}]$ so that $\sum_{\vec{x}} P(\vec{x};\vec{\lambda})\vec{\phi}(\vec{x}) = \vec{h}$, where $\vec{h}$ are observation data (with components indexed by μ). This can be expressed as finding the minimum of the convex energy function $\log Z[\lambda] - \vec{h}\cdot\vec{\lambda}$, where $Z[\vec{\lambda}] = \sum_{\vec{x}} e^{\vec{\lambda}\cdot\vec{\phi}(\vec{x})}$ is the partition problem. All problems of this type can be converted to a standard form where $\phi_\mu(\vec{x}) \geq 0,\ \forall\ \mu, \vec{x},\ h_\mu \geq 0,\ \forall\ \mu$, and $\sum_\mu \phi_\mu(\vec{x}) = 1,\ \forall\ \vec{x}$ and $\sum_\mu h_\mu = 1$ (Darroch and Ratcliff 1972). From now on we assume this form.

The GIS algorithm is given by

$$\lambda_\mu^{t+1} = \lambda_\mu^t - \log h_\mu^t + \log h_\mu,\ \forall\ \mu, \tag{25}$$

where $h_\mu^t = \sum_{\vec{x}} P(\vec{x};\vec{\lambda}^t)\phi_\mu(\vec{x})$. It is guaranteed to converge to the (unique) minimum of the energy function $\log Z[\lambda] - \vec{h}\cdot\vec{\lambda}$ and hence gives a solution to $\sum_{\vec{x}} P(\vec{x};\vec{\lambda})\vec{\phi}(\vec{x}) = \vec{h}$, (Darroch and Ratcliff 1972).

We now show that GIS can be reformulated as CCCP which gives a simple convergence proof of the algorithm.

Theorem 7. *We can express GIS as a CCCP algorithm in the variables* $\{r_\mu = e^{\lambda_\mu}\}$ *by decomposing the cost function* $E[\vec{r}]$ *into a convex term* $-\sum_\mu h_\mu \log r_\mu$ *and a concave term* $\log Z[\{\log r_\mu\}]$.

Proof. *Formulate the problem as finding the* $\vec{\lambda}$ *which minimizes* $\log Z[\lambda] - \vec{h}\cdot\vec{\lambda}$. *This is equivalent to minimizing the cost function* $E[\vec{r}] = \log Z[\{\log r_\mu\}] - \sum_\mu h_\mu \log r_\mu$ *with respect to* $\{r_\mu\}$ *where* $r_\mu = e^{\lambda_\mu},\ \forall\ \mu$. *Define* $E_{vex}[\vec{r}] = -\sum_\mu h_\mu \log r_\mu$ *and* $E_{cave}[\vec{r}] = \log Z[\{\log r_\mu\}]$. *It is straightforward to verify that* $E_{vex}[\vec{r}]$ *is a convex function (recall that* $h_\mu \geq 0,\ \forall\ \mu$*).*

To show that E_{cave} *is concave we compute its Hessian:*

$$\frac{\partial^2 E_{cave}}{\partial r_\mu \partial r_\nu} = \frac{-\delta_{\mu\nu}}{r_\nu^2}\sum_{\vec{x}} P(\vec{x} : \vec{r})\phi_\nu(\vec{x}) - \frac{\delta_{\mu\nu}}{r_\nu^2}\sum_{\vec{x}} P(\vec{x} : \vec{r})\phi_\nu(\vec{x})\phi_\mu(\vec{x})$$
$$-\frac{1}{r_\nu r_\mu}\{\sum_{\vec{x}} P(\vec{x} : \vec{r})\phi_\nu(\vec{x})\}\{\sum_{\vec{x}} P(\vec{x} : \vec{r})\phi_\mu(\vec{x})\}, \tag{26}$$

where $P(\vec{x};\vec{r}) = e^{\sum_\mu (\log r_\mu)\phi_\mu(\vec{x})}/Z[\{\log r_\mu\}]$.

The third term is clearly negative semi-definite. To show that the sum of the first two terms are negative semi-definite requires proving that

$$\sum_{\vec{x}} P(\vec{x}:\vec{r}) \sum_{\nu} (\zeta_\nu/r_\nu)^2 \phi_\nu(\vec{x}) \geq \sum_{\vec{x}} P(\vec{x}:\vec{r}) \sum_{\mu,\nu} (\zeta_\mu/r_\mu)(\zeta_\nu/r_\nu)\phi_\nu(\vec{x})\phi_\mu(\vec{x}) \tag{27}$$

for any set of $\{\zeta_\mu\}$. *This follows by applying the Cauchy-Schwarz inequality to the vectors* $\{(\zeta_\nu/r_\nu)\sqrt{\phi_\nu(\vec{x})}\}$ *and* $\{\sqrt{\phi_\nu(\vec{x})}\}$, *recalling that* $\sum_\mu \phi_\mu(\vec{x}) = 1, \ \forall \ \vec{x}$.

We now apply CCCP by setting $\frac{\partial}{\partial r_\nu} E_{vex}[r^{t+1}] = -\frac{\partial}{\partial r_\nu} E_{cave}[r^t]$. *We calculate* $\frac{\partial}{\partial r_\mu} E_{vex} = -h_\mu/r_\mu$ *and* $\frac{\partial}{\partial r_\mu} E_{cave} = (1/r_\mu)\sum_{\vec{x}} P(\vec{x}:\vec{r})\phi_\mu(\vec{x})$. *This gives:*

$$\frac{1}{r_\mu^{t+1}} = \frac{1}{r_\mu^t}\frac{1}{h_\mu} \sum_{\vec{x}} P(\vec{x}:\vec{r}^t)\phi_\nu(\vec{x}), \tag{28}$$

which is the GIS algorithm after setting $r_\mu = e^{\lambda_\mu}, \ \forall \ \mu$.

5.2 Sinkhorn's Algorithm

Sinkhorn's algorithm was designed to make matrices doubly stochastic (Sinkhorn 1964). We now show that it can reformulated as CCCP. In the next section we will described how Sinkhorn's algorithm, and variations of it, can be used to minimize convex energy functions such as those required for the inner loop of CCCP, see Theorem 3.

We first introduce Sinkhorn's algorithm (Sinkhorn 1964). Recall that an $n \times n$ matrix $\mathbf{\Theta}$ is a *doubly stochastic* matrix if all its rows and columns sum to 1. Matrices are *strictly positive* if all their elements are positive. Then Sinkhorn's theorem states:

Theorem (Sinkhorn 1964) *Given a strictly positive* $n \times n$ *matrix* $\mathbf{M}$, *there exists a unique doubly stochastic matrix* $\mathbf{\Theta} = \mathbf{EMD}$ *where* $\mathbf{D}$ *and* $\mathbf{E}$ *are strictly positive diagonal matrices (unique up to a scaling factor). Moreover, the iterative process of alternatively normalizing the rows and columns of* $\mathbf{M}$ *to each sum to* 1, *converges to* $\mathbf{\Theta}$.

Theorem 7. *Sinkhorn's algorithm is CCCP with a cost function* $E[r] = E_{vex}[r] + E_{cave}[r]$ *where*

$$E_{vex}[r] = -\sum_a \log r_a, \quad E_{cave}[r] = \sum_i \log \sum_a r_a M_{ia}, \tag{29}$$

where the $\{r_a\}$ *are the diagonal elements of* $\mathbf{E}$ *and the diagonal elements of* $\mathbf{D}$ *are given by* $\{\sum_a r_a M_{ia}\}$.

Proof. *It is straightforward to verify that Sinkhorn's algorithm is equivalent (Kosowsky and Yuille 1994) to minimizing an energy function* $\hat{E}[r,s] = -\sum_a \log r_a - \sum_i \log s_i + \sum_{ia} M_{ia} r_a s_i$ *with respect to* r *and* s *alternatively, where* $\{r_a\}$ *and* $\{s_i\}$ *are the diagonal elements of* $\mathbf{E}$ *and* $\mathbf{D}$. *We calculate* $E(r) = \hat{E}[r, s^*(r)]$ *where* $s^*(r) = \arg\min_s \hat{E}[r,s]$. *It is a direct calculation that* $E_{vex}[r]$ *is convex. The Hessian of* $E_{cave}[r]$ *can be calculated to be*

$$\frac{\partial^2}{\partial r_a \partial r_b} E_{cave}[r] = -\sum_i \frac{M_{ia} M_{ib}}{\{\sum_c r_c M_{ic}\}^2}, \tag{30}$$

which is negative semi-definite. The CCCP algorithm is

$$r_a^{t+1} = \sum_i \frac{M_{ia}}{\sum_c r_c^t M_{ic}}, \tag{31}$$

which is corresponds to one step of minimizing $\hat{E}[r,s]$ *with respect to* r *and* s, *and hence is equivalent to Sinkhorn's algorithm.*

5.3 Linear Constraints and Inner Loop

We now derive CCCP algorithms to minimize many of the update energy functions which can occur in the inner loop of CCCP algorithms. These algorithms are derived using similar techniques to those used by Kosowsky and Yuille (1994) to rederive Sinkhorn's algorithm, hence they can be considered to be generalizations of Sinkhorn. The linear assignment problem can also be solved using these methods (Kosowsky and Yuille 1994).

From Theorem 3, the update energy functions for the inner loop of CCCP are given by:

$$E(\vec{x};\vec{\lambda}) = \sum_i x_i \log x_i + \sum_i x_i a_i + \sum_\mu \lambda_\mu (\sum_i c_i^\mu x_i - \alpha_\mu). \tag{32}$$

Theorem 8. *Update energy functions, of form 32, can be minimized by a CCCP algorithm for the dual variables* $\{\lambda_\mu\}$ *provided the linear constraints satisfy the conditions* $\alpha_i \geq 0,\ \forall\ i$ *and* $c_i^\nu \geq 0,\ \forall\ i,\nu$. *The algorithm is:*

$$\frac{\alpha_\mu}{r_\mu^{t+1}} = e^{-1} \sum_i e^{-a_i} \frac{c_i^\mu}{r_\mu^t} e^{\sum_\nu c_i^\nu \log r_\nu}. \tag{33}$$

Proof. *First we scale the constraints we can require that* $\sum_\nu c_i^\nu \leq 1,\ \forall\ i$. *Then we calculate the (negative) dual energy function to be* $\hat{E}[\lambda] = -E[\vec{x}^*(\vec{\lambda}) : \vec{\lambda}]$ *where* $\vec{x}^*(\vec{\lambda}) = \arg\min_{\vec{x}} E[\vec{x};\vec{\lambda}]$. *It is straightforward to calculate:*

$$x_i^*(\vec{\lambda}) = e^{-1-a_i-\sum_\mu \lambda_\mu c_i^\mu},\ \forall\ i,\ \hat{E}[\vec{\lambda}] = \sum_i e^{-1-a_i-\sum_\mu \lambda_\mu c_i^\mu} + \sum_\mu \alpha_\mu \lambda_\mu. \tag{34}$$

To obtain CCCP, we set $\lambda_\mu = -\log r_\mu, \forall\ \mu$. *We set:*

$$E_{vex}[r] = -\sum_\mu \alpha_\mu \log r_\mu, \quad E_{cave}[r] = e^{-1} \sum_i e^{-a_i} e^{\sum_\mu c_i^\mu \log r_\mu}. \tag{35}$$

It is clear that $E_{vex}[r]$ *is a convex function. To verify that* $E_{cave}[r]$ *is concave we differentiate it twice:*

$$\frac{\partial E_{cave}}{\partial r_\nu} = e^{-1} \sum_i e^{-a_i} \frac{c_i^\nu}{r_\nu} e^{\sum_\mu c_i^\mu \log r_\mu},$$

$$\frac{\partial^2 E_{cave}}{\partial r_\nu \partial r_\tau} = -e^{-1} \sum_i e^{-a_i} \frac{c_i^\nu}{r_\nu} \delta_{\nu\tau} e^{\sum_\mu c_i^\mu \log r_\mu} + e^{-1} \sum_i e^{-a_i} \frac{c_i^\nu}{r_\nu} \frac{c_i^\tau}{r_\tau} e^{\sum_\mu c_i^\mu \log r_\mu}. \tag{36}$$

To ensure that this is negative semi-definite, it is sufficient to require that $\sum_\nu c_i^\nu x_v^2/r_\nu^2 \geq \{\sum_{nu} c_i^\nu x^\nu / r_\nu\}^2$ *for any set of* $\{x_\nu\}$. *This will always be true provided that* $c_i^\nu \geq 0,\ \forall\ i,\nu$ *and if* $\sum_\nu c_i^\nu \leq 1,\ \forall\ i$.

Applying CCCP to $E_{vex}[r]$ and $E_{cave}[r]$ gives the update algorithm.

An important special case of equation (32) is the energy function:

$$E_{eff}[S;p,q] = \sum_{ia} A_{ia} S_{ia} + \sum_{a} p_a (\sum_{i} S_{ia} - 1) + \sum_{i} q_i (\sum_{a} S_{ia} - 1) + 1/\beta \sum_{ia} S_{ia} \log S_{ia}, \quad (37)$$

where we have introduced a new parameter β.

As shown in Kosowsky and Yuille (1994), the minima of $E_{eff}[S;p,q]$ at sufficiently large β correspond to the solutions of the linear assignment problem whose goal is to select the permutation matrix $\{\prod_{ia}\}$ which minimizes the energy $E[\prod] = \sum_{ia} \prod_{ia} A_{ia}$, where $\{A_{ia}\}$ is a set of assignment values.

Moreover, the CCCP algorithm for this case is directly equivalent to Sinkhorn's algorithm once we identify $\{e^{-1-\beta A_{ia}}\}$ with the components of $\mathbf{M}$, $\{e^{-\beta p_a}\}$ with diagonal elements of $\mathbf{E}$, and $\{e^{-\beta q_i}\}$ with the diagonal elements of $\mathbf{D}$, see statement of Sinkhorn's theorem. Therefore Sinkhorn's algorithm can be used to solve the linear assignment problem (Kosowsky and Yuille 1994).

6 Conclusion

CCCP is a general principle for constructing discrete time iterative dynamical systems for almost any energy minimization problem. We have shown that many existing discrete time iterative algorithms can be re-interpreted in terms of CCCP. This includes EM algorithms, Legendre transforms, Sinkhorn's algorithm, and Generalized Iterative Scaling. Alternatively, CCCP can be seen as a special case of variational bounding, lower bound maximization, and surrogate functions. CCCP gives a novel geometrical way for understanding, and proving convergence of, existing algorithms.

Moreover, CCCP can also be used to construct novel algorithms. See, for example, recent work (Yuille 2002) where CCCP was used to construct a double loop algorithm to minimize

the Bethe/Kikuchi free energy (Yedidia, Freeman, and Weiss 2000).

There are interesting connections between our results and those known to mathematicians. After much of this work was done we obtained an unpublished technical report by D. Geman (1984) which states Theorem 2 and has results for a subclass of EM algorithms. There also appear to be similarities to the work of Hoang Tuy who has shown that any arbitrary closed set is the projection of a difference of two convex sets in a space with one more dimension. (See http://www.mai.liu.se/Opt/MPS/News/tuy.html). Byrne (2000) has also developed an *interior point algorithm* for minimizing convex cost functions which is equivalent to CCCP and which has been applied to image reconstruction.

Acknowledgements

We thank James Coughlan and Yair Weiss for helpful conversations. James Coughlan drew figure 1 and suggested the form of figure 2. Max Welling gave useful feedback on an early draft of this manuscript. We thank the National Institute of Health (NEI) for grant number RO1-EY 12691-01.

References

- Bregman, L.M. 1967. "The Relaxation Method of Finding the Common Point of Convex sets and its Application to the Solution of Problems in Convex Programming". *U.S.S.R. Computational Mathematics and Mathematical Physics.*, 7, 1, pp 200-217.

- Byrne, C. 2000. "Block-iterative interior point optimization methods for image reconstruction from limited data." *Inverse Problem.* **14**, pp 1455-1467.

- Darroch, J.N. and Ratcliff, D. 1972. "Generalized Iterative Scaling for Log-Linear

Models". *The Annals of Mathematical Statistics.* Vol. 43. No. 5, pp 1470-1480.

- Della Pietra, S., Della Pietra, V. and Lafferty, J. 1997. "Inducing features of random fields". *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 19(4), pp 1-13.

- Dempster, A.P., Laird, N.M. and Rubin, D.B. 1977. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the Royal Statistical Society, B, 39, 1-38.

- Durbin, R., Willshaw, D. 1987. "An Analogue Approach to the Travelling Salesman Problem Using an Elastic Net Method", Nature, v.326, n.16, , p.689.

- Durbin, R., Szeliski, R. and Yuille, A.L.. 1989." An Analysis of an Elastic net Approach to the Traveling Salesman Problem". *Neural Computation.* **1**, pp 348-358.

- Elfadel, I.M. 1995. "Convex potentials and their conjugates in analog mean-field optimization". *Neural Computation.* Volume 7. Number 5. pp. 1079-1104.

- Geman, D. 1984. "Parameter Estimation for Markov Random Fields with Hidden Variables and Experiments with the EM Algorithm". Working Paper no. 21. Dept. Mathematics and Statistics. University of Massachusetts at Amherst.

- Hathaway, R. 1986. "Another Interpretation of the EM Algorithm for Mixture Distributions". *Statistics and Probability Letters.* Vol. 4, pp 53-56.

- Hofmann, T. and Buhmann, J.M. 1997. "Pairwise Data Clustering by Deterministic Annealing". *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(1), 1-14.

- Jordan, M.I. and Jacobs, R.A. 1994. "Hierarchical mixtures of experts and the EM algorithm". *Neural Computation, 6*, 181–214.

- Jordan, M.I., Ghahramani, Z. and Jaakkola, T.S. and Saul, L.K. 1999 "An introduction to variational methods for graphical models". *Machine Learning 37*, pp 183-233.

- Kivinen, J. and Warmuth, M. 1997. "Additive versus Exponentiated Gradient Updates for Linear Prediction". *J. Inform. Comput.* 132 (1), pp 1-64.

- Kosowsky, J.J. and Yuille, A.L. 1994. "The Invisible Hand Algorithm: Solving the Assignment Problem with Statistical Physics". *Neural Networks.*, Vol. 7, No. 3, pp 477-490.

- Lange K, Hunter D.R., Yang, I. 2000, "Optimization transfer using surrogate objective functions" (with discussion), *Journal of Computational and Graphical Statistics*, 9: 1-59.

- Luttrell, S.P. 1994. "Partitioned mixture distributions: an adaptive bayesian network for low-level image processing." *IEEE Proceedings on Vision, Image and Signal Processing.* 141(4): 251-260.

- Marcus, C. and Westervelt, R.M. 1989. "Dynamics of Iterated-Mao Neural Networks". Physics Review A. **40**, pp 501-509.

- Minka, T.P. 1998. "Expectation-Maximization as lower bound maximization". Technical Report. Available from http://www.stat.cmu.edu/ minka/papers/learning.html.

- Mjolsness, E. and Garrett, C. 1990. "Algebraic Transformations of Objective Functions". *Neural Networks.* Vol. 3, pp 651-669.

- Neal, R.M. and Hinton, G.E. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Learning in Graphical Model M.I. Jordan (editor).

- Press, W.H., Flannery, B.R., Teukolsky, S.A. and Vetterling, W.T. 1986. **Numerical Recipes**. Cambridge University Press.

- Rangarajan, A., Gold, S. and Mjolsness, E. 1996. "A Novel Optimizing Network Architecture with Applications". *Neural Computation,* 8(5), pp 1041-1060.

- Rangarajan, A. Yuille, A.L. and Mjolsness, E. 1999. "A Convergence Proof for the Softassign Quadratic Assignment Algorithm". *Neural Computation.* **11**, pp 1455-1474.

- Rangarajan, A. 2000. "Self-annealing and self-annihilation: unifying deterministic annealing and relaxation labeling". *Pattern Recognition.* Vol. 33, pp 635-649.

- Rosenfeld, A., Hummel,R, and Zucker, S. 1976. "Scene Labelling by Relaxation Operations". *IEEE Trans. Systems MAn Cybernetic.* Vol 6(6), pp 420-433.

- Rustagi,J. 1976. **Variational Methods in Statistics**. Academic Press.

- Strang, G. 1986. **Introduction to Applied Mathematics**. Wellesley-Cambridge Press. Wellesley, Massachusetts.

- Sinkhorn, R. 1964. "A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices". *Ann. Math. Statist.*. 35, pp 876-879.

- Waugh, F.R. and Westervelt, R.M. 1993. "Analog neural networks with local competition: I. Dynamics and stability". *Physical Review E*, 47(6), pp 4524-4536.

- Yuille, A.L. 1990. "Generalized Deformable Models, Statistical Physics and Matching Problems," *Neural Computation*, **2** pp 1-24.

- Yuille, A.L. and Kosowsky, J.J. 1994. "Statistical Physics Algorithms that Converge." Neural Computation. **6**, pp 341-356.

- Yedidia, J.S., Freeman, W.T. and Weiss, Y. 2000. "Bethe free energy, Kikuchi approximations and belief propagation algorithms". Proceedings of NIPS'2000.

- Yuille, A.L. 2002. "A Double-Loop Algorithm to Minimize the Bethe and Kikuchi Free Energies". Neural Computation. In press.